AD-A120 317 CONSTRUCTION ENGINEERING RESEARCH LAB (ARMY) CHAMPAIGN IL F/G 9/2
SOFTWARE DOCUMENTATION FOR MILENG1/UTIL READ-ONLY-MEMORY MODULE--ETC(U)
SEP 82 L A THOMAS, J M DEPONAI
UNCLASSIFIED CERL-TR-P-134 NL

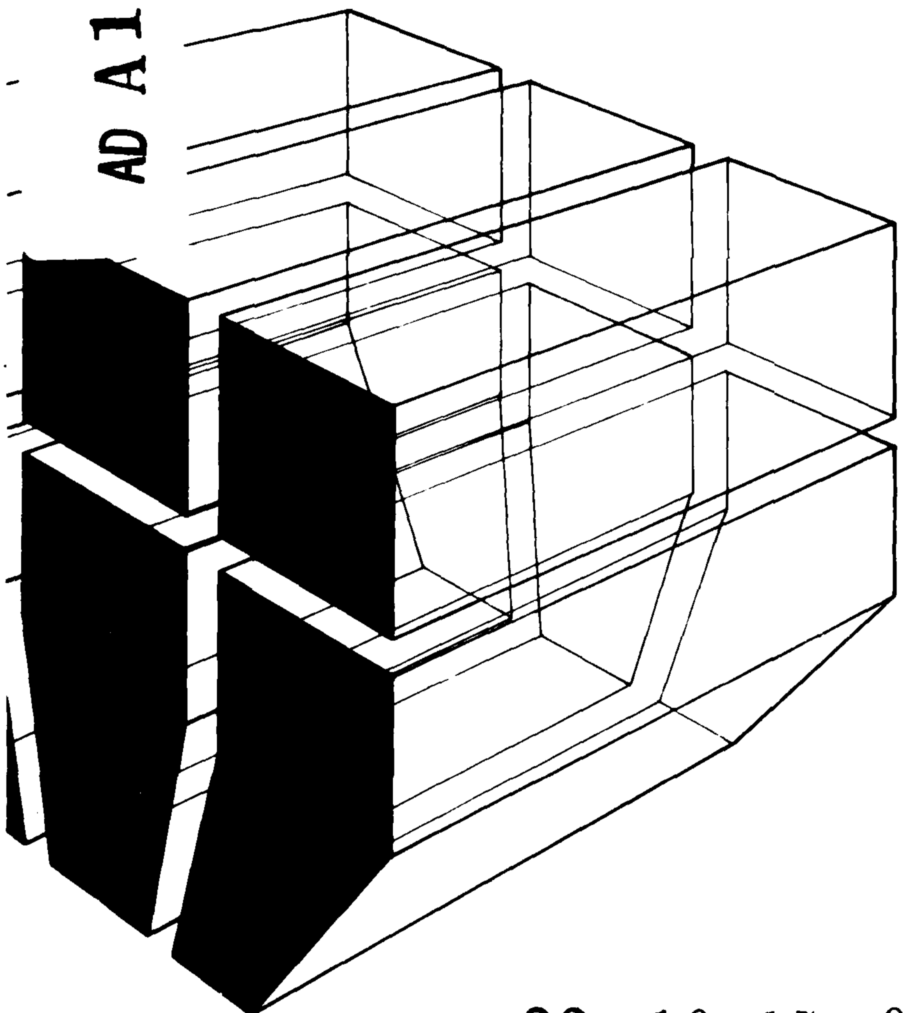
1 OF 2
AD A
120317

construction
engineering
research
laboratory

United States Army
Corps of Engineers
...Serving the Army
...Serving the Nation

TECHNICAL REPORT P-134
September 1982
Programmable Calculator Technology For Engineer Troop Units

SOFTWARE DOCUMENTATION FOR MILENG1/UTIL READ-ONLY-MEMORY MODULE



by
Laure A. Thomas
John M. Deponai III

CERL

Approved for public release; distribution unlimited.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CERL-TR-P-134	2. GOVT ACCESSION NO. AD-A120 317	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) SOFTWARE DOCUMENTATION FOR MILENG1/UTIL READ-ONLY-MEMORY MODULE		5. TYPE OF REPORT & PERIOD COVERED FINAL
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Laure A. Thomas John M. Deponai III		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS U.S. ARMY CONSTRUCTION ENGINEERING RESEARCH LABORATORY P.O. BOX 4005, CHAMPAIGN, IL 61820		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 4A762731AT41-D-049
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE September 1982
		13. NUMBER OF PAGES 174
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Copies are obtainable from the National Technical Information Service
Springfield, VA 22151

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

HP 41c
military engineering
computer applications
programmable calculators

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report documents six pilot military engineering programs and ten utility routines developed as part of an ongoing study to evaluate whether programmable hand-held calculators can increase the efficiency or military effectiveness of engineer troop units from platoon through brigade levels.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

FOREWORD

This investigation was conducted for the Directorate of Military Programs. Office of the Chief of Engineers (OCE), under Project 4A762731AT41, "Design, Construction, and Operation and Maintenance Technology for Military Facilities"; Task D, "Combat Engineering Strategy"; Work Unit 049, "Programmable Calculator Technology for Engineer Troop Units." The applicable STO is 81-5.1:19. The OCE Technical Monitors were LTC John Howard and Dr. Clemens Meyer, both of DAEN-ZCM.

This work was performed by the Facility Systems Division (FS) of the U.S. Army Construction Engineering Research Laboratory (CERL). Mr. E. A. Lotz is Chief of CERL-FS.

The cooperation and contributions of many persons on the staff of the U.S. Army Engineer School are gratefully acknowledged. CPT Scott Loomer of the Defense Mapping School is especially commended for developing the Bridge Classification Program and all but two of the global utility routines described in this report.

COL Louis J. Circeo is Commander and Director of CERL, and Dr. L. R. Shaffer is Technical Director.

CONTENTS

TABLES

FIGURES

FIGURES (Cont'd)

FIGURES (Cont'd)

SOFTWARE DOCUMENTATION FOR MILENG1/UTIL
READ-ONLY-MEMORY MODULE

1 INTRODUCTION

Background

Recent advancements in the state of the art of programmable calculators have indicated that these devices might help military engineers work more efficiently. To determine the potential of these systems, in March 1980 the U.S. Army Engineer School asked the U.S. Army Construction Engineering Research Laboratory (CERL) to see how hand-held programmable calculators could be exploited by combat engineers.

Purpose

The overall objective of the research is to determine whether hand-held programmable calculators can increase the efficiency or military effectiveness of engineer troop units from platoon through brigade levels. The objective of the first phase of the study was to develop a set of pilot military engineering programs that could be stored on a read-only-memory (ROM) module (MILENG1/UTIL). This report provides documentation for the six pilot programs and 10 utility routines produced to date. A future report will describe the overall results of the study.

Approach

To date, CERL has accomplished the following steps in pursuit of the research objective.

1. A state-of-the-art calculator system -- the Hewlett Packard (HP)-41c — was selected because of the size of its memory (four 8000-byte ROMs) and its alphanumeric display capability.

2. A survey of all active Army engineer units was made to establish priorities among the functional needs of combat engineer field troops.

3. Software programs were written to address the needs with the highest priority.

4. Magnetic card versions of these programs were field tested for 5 months in eight engineer troop units in Korea, the United States, and Europe. Based on the results of these field tests and on programming improvements developed at the U.S. Army Engineer School and at the Defense Mapping School, the programs were revised once more.

5. The MILENG1/UTIL ROM pilot module was developed and documented (in this report) for the most useful programs and utility routines.

Future activity in support of the research will include:

1. Distribution of the MILENG1/UTIL ROM to field test units and to the USAES for testing;

2. Evaluation of the test results;

3. Writing a report documenting the conduct of the research and making recommendations on use of programmable calculators by engineer troops.

Outline of This Report

Chapter 2 is an overview of the scope of the six major application programs, the 10 global utility routines, and the various conventions used in the programs on the MILENG1/UTIL module. Comprehensive descriptions of each program and of the global utility routines are presented in Appendices A through G. Appendix H provides a set of blank forms that can be used to document future user-developed programs in a standard format.

Mode of Technology Transfer

It is recommended that the technology transfer media for use of programm able calculators by engineer troops include a Department of the Army field manual and on-site training courses on the use of the MILENG1/UTIL ROM module.

2 MILENG1/UTIL PROGRAMS, ROUTINES, AND CONVENTIONS

Major Application Programs

The MILENG1/UTIL ROM module stores six major application programs designed to be user-friendly, yet efficient in terms of execution time required.

The Bridge Classification Program

The bridge classification program is used with certain tables in Field Manual (FM) 5-34 to help the user determine bridge superstructure classification.[1] This program may be used for both timber and steel stringer bridges. It first asks the user for information such as the bridge's basic dimensions. It then refers the user to appropriate tables and figures in FM 5-34, tells him what entry values he needs, and asks him for the value of the variables corresponding to those entry conditions. The user extracts the appropriate values from the manual and inputs them to the calculator. The program determines (1) the limiting classifications for one- and two-way traffic by wheeled and by track vehicles; (2) the constraints imposed by moment and shear capacity, deck thickness, and roadway width; and (3) the need for additional braces. A more detailed description is in Appendix A.

The CPM Program

The critical path method (CPM) program provides an easy way to do the tedious calculations associated with using CPM project control, a management method becoming common in the Army. The CPM program uses "activity on the node" logic. Up to 98 activities can be analyzed if the full resident capacity for data storage of the HP-41cv model is used; on the other hand, up to 20 activities can be done on the HP-41c model without memory modules.

The program computes the total float, the early start and finish times, and the late start and finish times for each activity. When the HP-41c is used with a printer, boxes for the various CPM activities are printed. The user can cut these out, paste them up, and graph logical relationships. When no printer is available, the user must write down the output; then diagrams can be drawn by hand and annotated with the correct information. A detailed description of the CPM program is in Appendix B.

Road Crater Program

The road crater program computes the weight of explosives, number of cratering charges, and number and depth of holes needed to produce hasty, deliberate, or relieved face road craters of user-specified lengths. A complete description of the program is in Appendix C.

---

[1] Engineer Field Data, Field Manual (FM) 5-34 (Headquarters [HQ], Department of the Army [DA], September 1976).

*In this report, the masculine pronoun is used to refer to both sexes.

The Demolition Program

The demolition program addresses three common engineer activities: cutting timber and steel, and breaching walls. A menu of explosive types to be used is presented. The program has three timber cutting options: internal and external charge placement, and abatis. The steel cutting options cover railroad rails, round steel sections, structural steel sections, and carbon steel rods. The breaching applications are used with the applicable tables in FM 5-34. A complete description is in Appendix D.

The Minefield Program

The minefield program computes the logistical requirements for installing a standard pattern minefield given the field density, irregular outer-edge cluster composition, field length and depth, and conditions under which the work is to be done. The program is detailed in Appendix E.

The Wire Obstacle Program

The wire obstacle program computes the logistical requirements for installing any of seven common wire obstacles: double apron fence, four- and two-pace; double apron fence, six- and three-pace; high wire; low wire; four-strand fence; triple standard concertina; and general purpose barbed tape obstacle. The program can also be used to compute the effective length of the obstacle according to its function and location on the battlefield. If the user already knows the effective length, he can input the effective length directly. A complete description is in Appendix F.

Global Utility Routines

The "friendliness" of the application programs results largely from a set of global utility routines providing standard input/output interfaces to users. Using these routines lets a programmer communicate with users in a standard format and reduces the number of formats with which users must be familiar.

The 10 "global subroutines" are used by the six main application programs, and by each other in some cases. Assuming a MILENG1/UTIL module is plugged in the calculator, these utility routines are conveniently available for field programmers to use in writing ad hoc programs. Wide-spread adoption of these routines would help standardize the input/output formats that military users have to be familiar with.

The function of each of the 10 global subroutines is described briefly below. A more detailed description is given in Appendix G.

**S* - Insures that enough data registers are available to run a given program; requires use of routines *0 and *D.

**I* - Provides for input of numerical data; requires use of routines *0 and *D; rejects alpha input attempts and repeats question.

2 MILENG1/UTIL PROGRAMS, ROUTINES, AND CONVENTIONS

Major Application Programs

The MILENG1/UTIL ROM module stores six major application programs designed to be user-friendly, yet efficient in terms of execution time required.

The Bridge Classification Program

The bridge classification program is used with certain tables in Field Manual (FM) 5-34 to help the user determine bridge superstructure classification.[1] This program may be used for both timber and steel stringer bridges. It first asks the user for information such as the bridge's basic dimensions. It then refers the user to appropriate tables and figures in FM 5-34, tells him what entry values he needs, and asks him for the value of the variables corresponding to those entry conditions. The user extracts the appropriate values from the manual and inputs them to the calculator. The program determines (1) the limiting classifications for one- and two-way traffic by wheeled and by track vehicles; (2) the constraints imposed by moment and shear capacity, deck thickness, and roadway width; and (3) the need for additional braces. A more detailed description is in Appendix A.

The CPM Program

The critical path method (CPM) program provides an easy way to do the tedious calculations associated with using CPM project control, a management method becoming common in the Army. The CPM program uses "activity on the node" logic. Up to 98 activities can be analyzed if the full resident capacity for data storage of the HP-41cv model is used; on the other hand, up to 20 activities can be done on the HP-41c model without memory modules.

The program computes the total float, the early start and finish times, and the late start and finish times for each activity. When the HP-41c is used with a printer, boxes for the various CPM activities are printed. The user can cut these out, paste them up, and graph logical relationships. When no printer is available, the user must write down the output; then diagrams can be drawn by hand and annotated with the correct information. A detailed description of the CPM program is in Appendix B.

Road Crater Program

The road crater program computes the weight of explosives, number of cratering charges, and number and depth of holes needed to produce hasty, deliberate, or relieved face road craters of user-specified lengths. A complete description of the program is in Appendix C.

---

[1] Engineer Field Data, Field Manual (FM) 5-34 (Headquarters [HQ], Department of the Army [DA], September 1976).

*In this report, the masculine pronoun is used to refer to both sexes.

*O - Outputs numerical answer in alpha form; requires use of routine *D.

*D - Displays output and alerts user by sounding a two-tone signal.

*Y - Appends a "(Y/N)?" to yes/no questions; sets flag "10" if response is "Y"; clears flag "10" if response is "N"; rejects all other responses as invalid, then repeats original question.

*A - Provides for input of alpha data; rejects numeric responses and repeats original question.

*F - Clears general purpose flags "01 through 07".

*C - Clears a block of registers specified by the programmer.

*R - Provides "round-up" capability.

*P - Displays "END PROGRAM" message and sounds a two-tone signal to announce the message.

Input Conventions

A user has to learn only a few conventions to be able to use the MILENG1/UTIL programs effectively.

The Yes/No Request

The user is presented with an alpha string that ends with "(Y/N)?". The user must respond with either a "Y R/S"* (for yes), or "N R/S" (for no). Any other responses will be rejected by the program and the questions will be displayed again. When the program presents the user such a question, the calculator is automatically put in the alpha mode while awaiting the user's response.

The Numeric Input Request

The user is presented with an alpha string representing the variable's name and unit of measure, ending with "=?". The user must key in a numeric string and press the R/S key. The ENTER key is not used. If an alpha string is keyed in, it is rejected and the question repeated. If the input is not within the allowable range specified in the program, one of the following messages is displayed: "MUST BE <= (some maximum value)" or "MUST BE >= (some minimum value)." Then the original question is repeated.

Users also should be aware of an optional format for a numerical input request. This option is not used in any of the application programs on the MILENG1/UTIL module. However, in future applications, some programmers might exercise the "FS 09" option of the global *I subroutine (described in

---

* When several letters/symbols are underlined, it means that as a group they identify the name of a single key on the calculator. Only key function names are shown. Use of the shift key, if needed, is assumed. Single letters or numbers represent individual keys on the calculator.

Appendix G). The format is much the same as described above. However, a tentative numeric answer would be displayed between the "=" and the "?". If the user agrees with the tentative answer, he would press only the R/S key and the program would use that value. If the user wants to use another value of the variable, he would key in the numeric string, then press the "R/S" key. This option should not be confused with the yes/no format described above and is distinguished by the fact that no "(Y/N)" appears in the question and the calculator is not left in the "alpha" mode. However, if the user inadvertently answers with a "Y R/S" or "N R/S" response, the subroutine simply rejects that response and repeats the original question.

The Alpha Input Request

The user is presented with an alpha string ending with "=?" and the calculator is left in the "alpha" mode while it awaits a response. This feature is not used by any of the application programs presently on the MILENG1/UTIL module but could appear in future program applications. The user must key in the alpha string answer, then press the R/S key. The ENTER key is not used. If the user were to take the calculator out of the alpha mode and input a numeric answer, the program would reject the answer and repeat the question. Only alpha characters are accepted when this input feature is used.

Users should be aware of an optional format for an alpha input request. It is not used in MILENG1/UTIL programs, but may be used in future applications if programmers exercise the "FS 09" option of the global *A subroutine (described in Appendix G). The format is the same as above, but a tentative alpha answer appears between the "=" and the "?". If the user agrees with this, he presses the R/S key and the program uses that answer. If the user wants to try another response, he would key in the alpha string, then press the R/S key. However, if the user answers with a "Y R/S" or a "N R/S" response (to signify that he agreed or disagreed with the tentative response), the "Y" or "N" would be misinterpreted by the program as a replacement value for the tentative alpha-string. Users should be careful to answer "Y" or "N" only to questions that are displayed with a "(Y/N)" immediately preceding the question mark.

Output Conventions

All "MILENG1/UTIL" programs are designed to run with or without a printer. If a printer is attached, the program stops only when the user must respond to a request for input and at the very end of the program. If a printer is not attached, the program will stop at these same points and after each line of output. To continue with the program execution and output operations the user must press the R/S key each time the program stops. If a printer is used, it should be set to the "NORM" mode.

Miscellaneous Conventions

Program Access

To access a particular program or global subroutine, the user "executes" the program name. The names of the six application programs are "BRDGCLS," "CPM," "CRATER," "DEMO," "MINES," and "WIRE"; the 10 global subroutines are *S, *I, *O, *D, *Y, *A, *C, *R, *P, and *F. For example, to call the MINES program, the user would press "XEQ ALPHA MINES ALPHA".

Size Check

The first step in each application program on the MILENG1/UTIL module is to insure that there are enough data registers available to execute the program. If there are, the program continues; if not, the program tells the user to "RESIZE > (No. of Registers Required, minus one)." Note that the 00 register counts as one of the registers -- thus, the "minus one." The minimum number of registers required for each program is as follows:

Program	No. of Registers Required
BRDGCLS	52
CPM	(No. of Activities x 2) + 43
CRATER	40
DEMO	41
MINES	53
WIRE	44

Register Use

Registers 00 through 19 are reserved for the user. The contents of these registers are not altered by the programs on MILENG1/UTIL. The contents of registers 20 and above may be altered, depending on which programs are executed.

Programming Conventions

All of the programs on "MILENG1/UTIL" use the programming conventions described below. For the sake of uniformity, it is recommended that future programs use the same conventions.

1. General purpose flags "00 through 07" and registers 30 and beyond used in a program are cleared at the start of the program to initialize, and at the end to "tidy up."

2. The date on which the last change to a program was made is recorded just before the "END" statement. This is especially useful when programs are being developed and there are many copies that differ only in a few details. One can easily tell whether the latest version is being used.

3. When programming for sequential input, one should not assume that values entered previously are still in the stack. Users can too easily alter the contents of the stack during the input/output process. For example, a

user might erroneously press the <u>ENTER</u> key before the <u>R/S</u> key when inputting data. This will not invalidate the particular input being done, but it will change the contents in the stack. Another possibility is that intermediate output values might be used to compute other requirements; this would also alter the stack contents.

4. As much as possible, data required for the program to execute are requested at the start of the program, not at the point in the program where input is needed. This is more convenient for the user.

5. Data registers 00 through 19 are reserved for users. Registers 20 through 29 are reserved for use by global subroutines. Registers 30 and beyond are available for data storage needed to run the application programs.

6. When menus are presented by the program, a loop is provided back to the beginning of the menu if the user does not choose one of the items. This enables the user to see what he has to choose from before making his decision.

Documentation Conventions

The forms in Appendix H can be used to document future programs that may be written by military engineers. Appendices A through G provide specific examples of the level of detail that might be employed to describe the programs.

APPENDIX A:
"BRDGCLS" PROGRAM DETAILS

This appendix provides detailed information about the bridge classification ("BRDGCLS") program. Figure A1 shows the typical sequence of events and the options that a user encounters when executing this program. Table A1 is a sample problem showing the specific steps that must be followed when one uses the HP-41 calculator without a printer attached. In this example, the classification of a timber trestle bridge is determined.

"BRDGCLS" requires the use of tables/figures in FM 5-34. In step 19 of Table A1, for example, the user must input the moment capacity of a timber stringer from Table 7-1, FM 5-34. In any step that displays a message, such as step 2 in the same example, the user has to press the R/S key to restart the program after each part of the message is displayed. If a printer were attached and set to the "NORM" printer mode, the program would advance automatically. Steps 46 through 51 of Table A1 and all other steps that output data would also be executed automatically if a printer were attached. Figure A2 shows the output from a run of the "BRDGCLS" program with a printer attached.

Abbreviations used in "BRDGCLS" are:

Symbol	Meaning
CLS,CLASS	Classification
DT	Deck thickness
FIG	Figure
FT	Feet
IN	Inches
KP	Kip
L,M	Maximum span length
LAM	Laminated
LN	Lane
M	Moment capacity
M,DL	Dead load moment
M,LL	Live load moment
N1	Effective number of stringers/lane
N2	Effective number of stringers/lane for a two-lane bridge
S,B	Maximum bracing spacing
S,S	Stringer spacing
STR	Stringer
TAB	Table
TBR	Timber
THICK	Thickness
V	Shear capacity
V,DL	Dead load shear
V,LL	Live load shear
WY	Way
(Y/N)	(Yes/No)
#	Number

There are several operating limits for input variables in the program:

Variable	Units	Minimum	Maximum
Road width	Feet	8	50
Span length	Feet	10	200
Stringer number	Each	2	25
Stringer width	Inches	4	20
Stringer depth	Inches	6	60
Stringer flange thickness	Inches	0.3	2
Stringer spacing	Inches	10	100
Deck thickness	Inches	2	12
% laminated	Percent	0	100
Number of braces	Each	0	20
Moment capacity	Kip-feet	3	3100
Shear capacity	Kip	3	600
Maximum span length	Feet	9	135
Maximum bracing spacing	Feet	6	26
Dead load moment	Kip-feet	3	1200
Dead load shear	Kip	1	65
Wheel classification	Class	0	150
Track classification	Class	0	150
Classification	Class	0	150

Algorithms used in the program were taken from FM 5-34 (pp 170-184 and 199-203). Critical assumptions and formulas are described below:

1. Conduct a bridge reconnaisance to obtain the following information on the existing bridge:

Road width (W_R) in feet
Span length (L) in feet
Type, size, and number of stringers
Thickness of decking in inches and type of decking
Number of lateral braces.

2. In Table 7-1 or 7-2 of FM 5-34, locate the stringer to be classified and determine the moment capacity (M) in kip-feet, shear capacity (V) in kips, maximum span length (L,M) in feet, and maximum bracing spacing (S,B) in feet (for steel stringers).

3. Determine if bridge is one or two lanes. If $W_R \geq 18$ ft, the bridge is two-lane.

4. Obtain the dead load moment per span (M,DL) in kip-feet and dead load shear per span (V,DL) in kips for the type of superstructure involved from Figure 7-4, FM 5-34.

5. Calculate number of lateral braces (N_b):

Steel: $N_b = \frac{L}{S,B} + 1$

Timber: $N_b = 3$, If $d > 2b$ for stringer.

If necessary, add bracing as required.

6. Calculate dead load moment per str' ıger (M,DL/STR):

$$\frac{M,DL}{STR} = \frac{M,DL}{N_s} \text{ (ft-kip) where } N_s = \text{number of stringers.}$$

7. Calculate live load moment per stringer (M,LL/STR):

$$\text{Steel: } \frac{M,LL}{STR} = \frac{M-M,DL/STR}{1.15} \text{ (ft-kip).}$$

$$\text{Timber: } \frac{M,LL}{STR} = M-(M,DL/STR) \text{ (ft-kip).}$$

Check the maximum span length (L,M) of the stringer from Table 7-1 or 7-2, FM 5-34. If L,M > L, proceed to Step 8. If L,M < L, multiply M,LL/STR by the ratio L,M/L to obtain a new, lower value of M,LL/STR.

8. Calculate effective number of stringers per lane for one-way (N_1) and two-way traffic (N_2).

One-way traffic: $N_1 = \frac{5}{S,S} + 1$ where S,S = stringer spacing in inches

Two-way traffic: $N_2 = \frac{3}{8} \times N_s$

(Note: Do not round off N_1 or N_2)

For one-way: If $N_1 > N_2$ use N_1
If $N_2 > N_1$ use N_1

For two-way: If $N_1 > N_2$ use N_2
If $N_2 > N_1$ use N_1.

9. Calculate live load moment per lane (M,LL):

$M,LL = N_1 \times (M,LL/STR)$ (ft-kip/lane).

10. Determine the bridge classification based on bending moment by entering M,LL and span length into Figure 7-3, FM 5-34, for both wheeled and tracked vehicles. Note: If $N_1 > N_2$, return to Step 9 and calculate M,LL using N_2 in place of N_1. Obtain another classification for two-way traffic from Figure 7-3, FM 5-34.

11. Calculate dead load shear per stringer (V,DL/STR):

$$\frac{V,DL}{STR} = \frac{V,DL}{N_S} \text{ (ft-kip).}$$

12. Calculate live load shear per stringer (V,LL/STR):

$$\frac{V,LL}{STR} = V - \frac{V,DL}{STR} \text{ (ft-kip).}$$

13. Calculate live load shear per lane (V,LL).

For steel: $V,LL = \frac{2 \times (V,LL/STR)}{1.15}$ (kip/lane).

For timber: $V,LL = \frac{16}{3} \times \frac{V,LL}{STR} \times \frac{N_1}{N_1 + 1}$ (kip/lane).

14. Determine the bridge classification based on shear by entering V,LL and span length into Figure 7-5, FM 5-34, for both wheeled and tracked vehicles. Note: If $N_1 > N_2$, return to Step 13 and calculate V,LL using N_2 in place of N_1. Obtain another classification for two-way traffic from Figure 7-5, FM 5-34.

15. In Table 7-6, FM 5-34, find the maximum classification based on roadway width restrictions for one-way and two-way traffic.

16. Determine the decking classification:

a. Find the effective thickness for the type of decking (t_{eff}) involved:

For laminated decking: $t_{eff} = (t_{actual})$(% laminated)

For plank deck: $t_{eff} = t_{actual} - 2$ in.

(Note: Subtract 2 in. for multilayered plank deck only.)

b. Find the decking class by entering the effective thickness and stringer spacing into Figure 7-7, FM 5-34.

17. Find the final bridge classification by comparing the classes for moment, shear, two-way width, and deck, and then selecting the lowest critical class for each type crossing, wheeled or track.

"BRDGCLS" uses registers 30 through 51 to store the values described in Table A2. The program uses general purpose flags 00 through 05 as described below.

Flag	Usage
00	Set if steel stringers are used
01	Set if a laminated deck
02	Set if a two-lane bridge
03	Set if $N_1 > N_2$
04	Set if additional braces are needed
05	Temporary use -- intermediate computation

Table A3 describes the general function of each part of the program, by label. Figure A3 is a label wiring diagram showing how the different parts of the program relate to each other. A circular loop on the diagram indicates a return to the same label. A two-headed arrow pointing to and from a subroutine indicates that the program executes that as a local subroutine then

returns to the main program. Global subroutines, used by the major application programs on the MILENG1/UTIL module, are not shown on the wiring diagram. Global subroutines are described separately in Appendix G.

Figure A4 is a detailed flowchart of the "BRDGCLS" program, and Figure A5 lists the program steps.

User executes "BRDGCLS"

User enters road width in feet

User enters span length in feet

User enters number of stringers

User decides if stringers are steel

User enters stringer width in inches

User enters stringer depth in inches

User enters stringer flange thickness in inches*

User enters stringer spacing in inches

User enters deck thickness in inches

User decides if laminated

(No) | (Yes)

User decides if layered | User enters % laminated

User enters number of braces

User enters moment capacity (kip-ft) from Tab. 7-1 or 7-2, FM 5-34

User enters shear capacity (kips) from Tab. 7-1 or 7-2, FM 5-34

Figure A1. "BRDGCLS" program sequence of events.

↓

User enters maximum span length (ft) from Tab. 7-1 or 7-2, FM 5-34

↓

User enters maximum bracing spacing (ft) from Tab. 7-2. FM 5-34*

↓

User enters dead load moment (kip-ft) from Fig. 7-4, FM 5-34

↓

User enters dead load shear (kip) from Fig. 7-4, FM 5-34

↓

Program computes/outputs dead load moment per stringer, live load moment per stringer, effective number of stringers per lane (N_1), and effective number of stringers per lane for a two-lane bridge (N_2)

↓

User enters class of wheeled vehicle from Fig. 7-3, FM 5-34

↓

User enters class of tracked vehicle from Fig. 7-3, FM 5-34

↓

Program computes/outputs dead load shear per stringer and live load shear per stringer

↓

User enters class of wheeled vehicle from Fig. 7-5, FM 5-34

↓

User enters class of tracked vehicle from Fig. 7-5. FM 5-34

↓

Program computes/outputs width classification

↓

User enters decking classification from Fig. 7-7, FM 5-34

↓

Program computes/outputs
Final classifications

*Only if steel stringers are used

Figure A1. (Cont'd).

```
        XEQ "BRDGCLS"
BRIDGE CLASS
RECON
ROAD WIDTH(FT)=?
              23.      RUN
SPAN LENGTH(FT)=?
              17.      RUN
STRINGERS:
STR, NUMBER=?
               9.      RUN
STEEL(Y/N)?
N                      RUN
STR. WIDTH(IN)=?
               8.      RUN
STR. DEPTH(IN)=?
              18.      RUN
STR. SPACING(IN)=?
              35.      RUN
DECK THICK.(IN)=?
               6.      RUN
LAMINATED(Y/N)?
N                      RUN
DECK LAYERED(Y/N)?
Y                      RUN
#BRACES=?
               2.      RUN
CLASS:
TAB.7-1,FM5-34:
8.0X81.0
M(KP-FT)=?
            86.40      RUN
V(KIP)=?
            14.40      RUN
L,M(FT)=?
            21.50      RUN
FIG.7-4,FM5-34.
TBR,2 LN,17.FT:
M,DL(KP-FT)=?
            37.36      RUN
V,DL(KIP)=?
             8.75      RUN
```

```
M,DL/STR=4.2
M,LL/STR=82.2
N1=2.71
N2=3.38
FIG.7-3,FM5-34:
17.FT.&223.M,LL:
CLS, WHEEL=?
           60.         RUN
CLS, TRACK=?
           40.         RUN
V,DL/STR=1.0
V,LL/STR=13.4
FIG.7-5,FM5-34:
17.FT.&52.V,LL:
CLS, WHEEL=?
           50.         RUN
CLS, TRACK=?
           40.         RUN
WIDTH CLS:
1 WAY=100.
2 WAY=30.
FIG.7-7,FM5-34:
DT=4.0&S,S=35.:
CLASS=?
           12.         RUN
FINAL CLASS:
ADD 1. BRACES
1WY WHEEL=12.
2WY WHEEL=12.
1WY TRACK=12.
2WY TRACK=12.
END PROGRAM
```

Figure A2. "BRDGCLS" program example -- with printer.

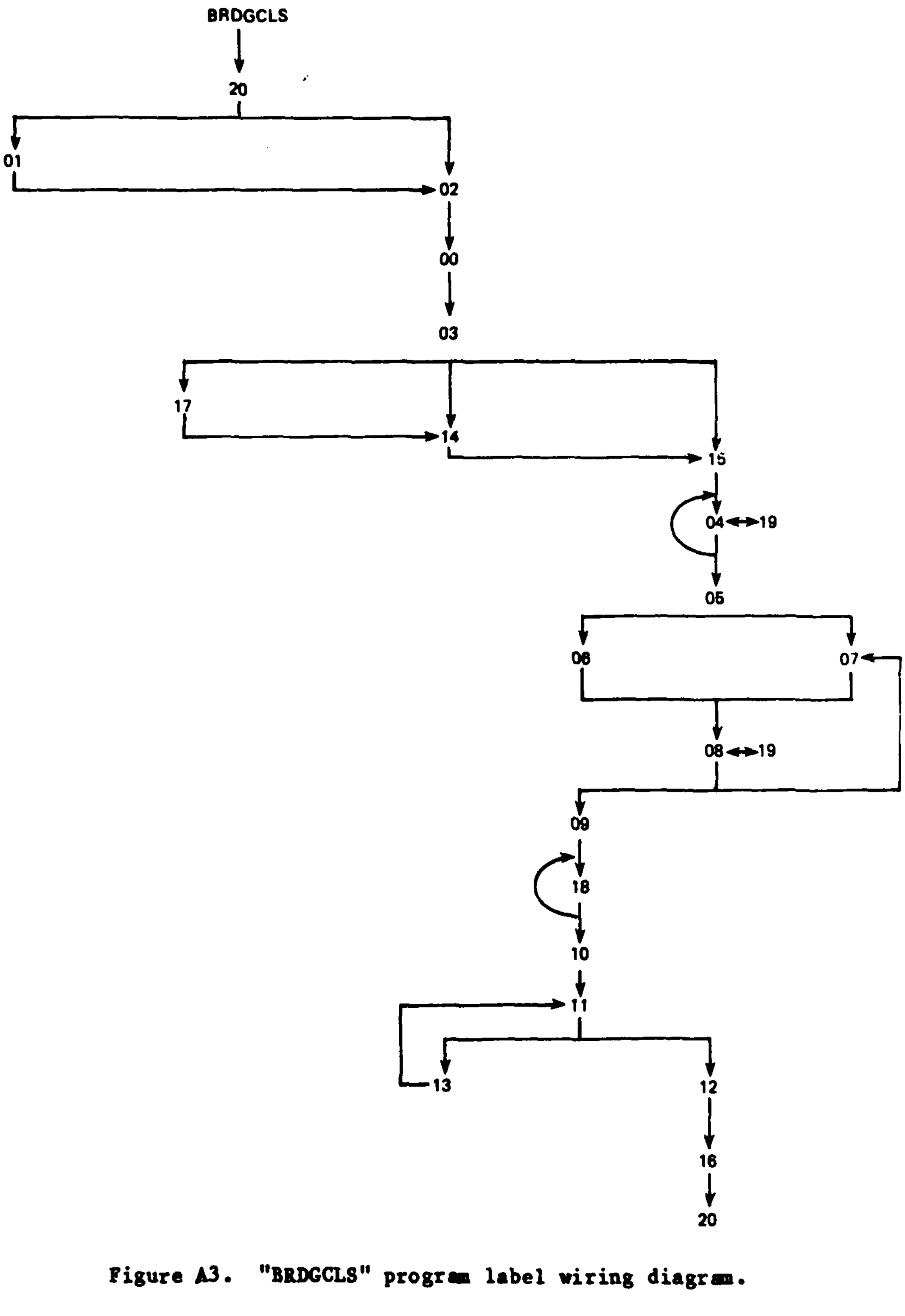


Figure A3. "BRDGCLS" program label wiring diagram.

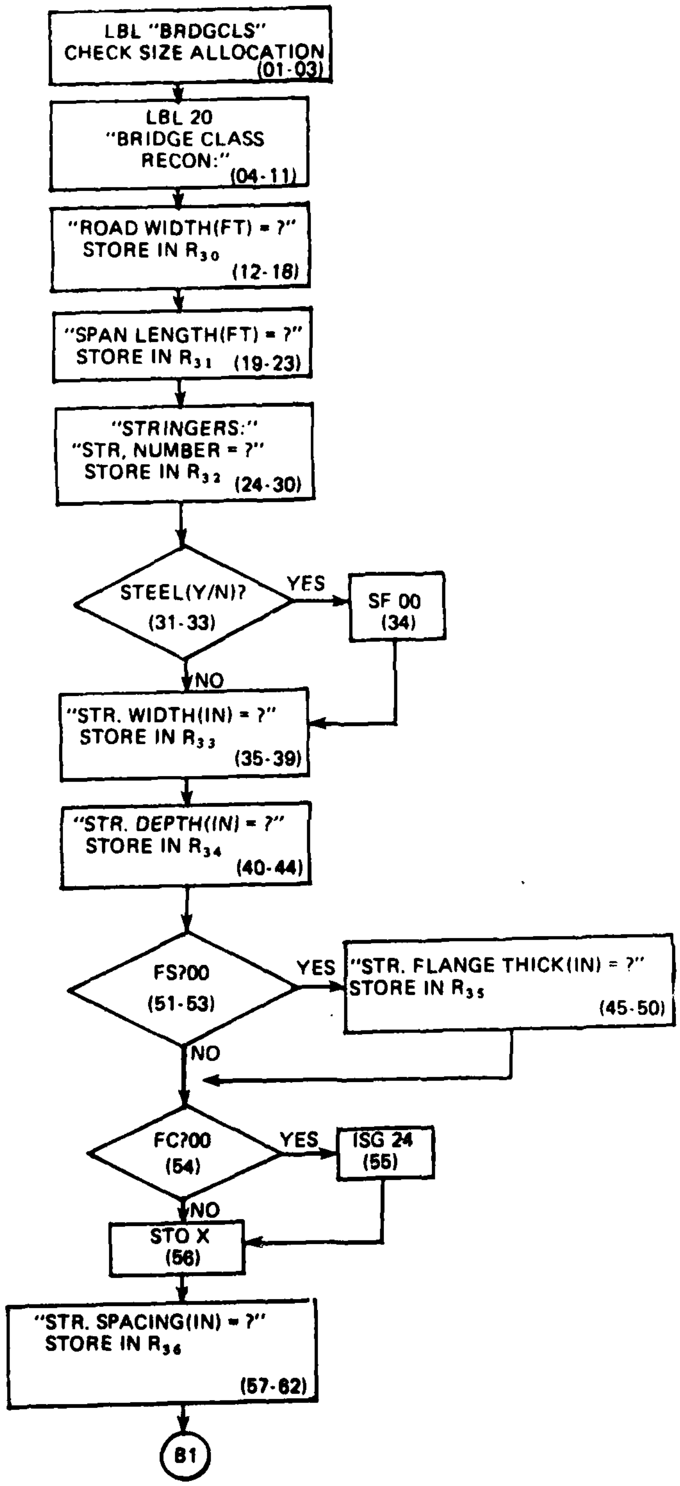


Figure A4. "BRDGCLS" program -- detailed flowchart.

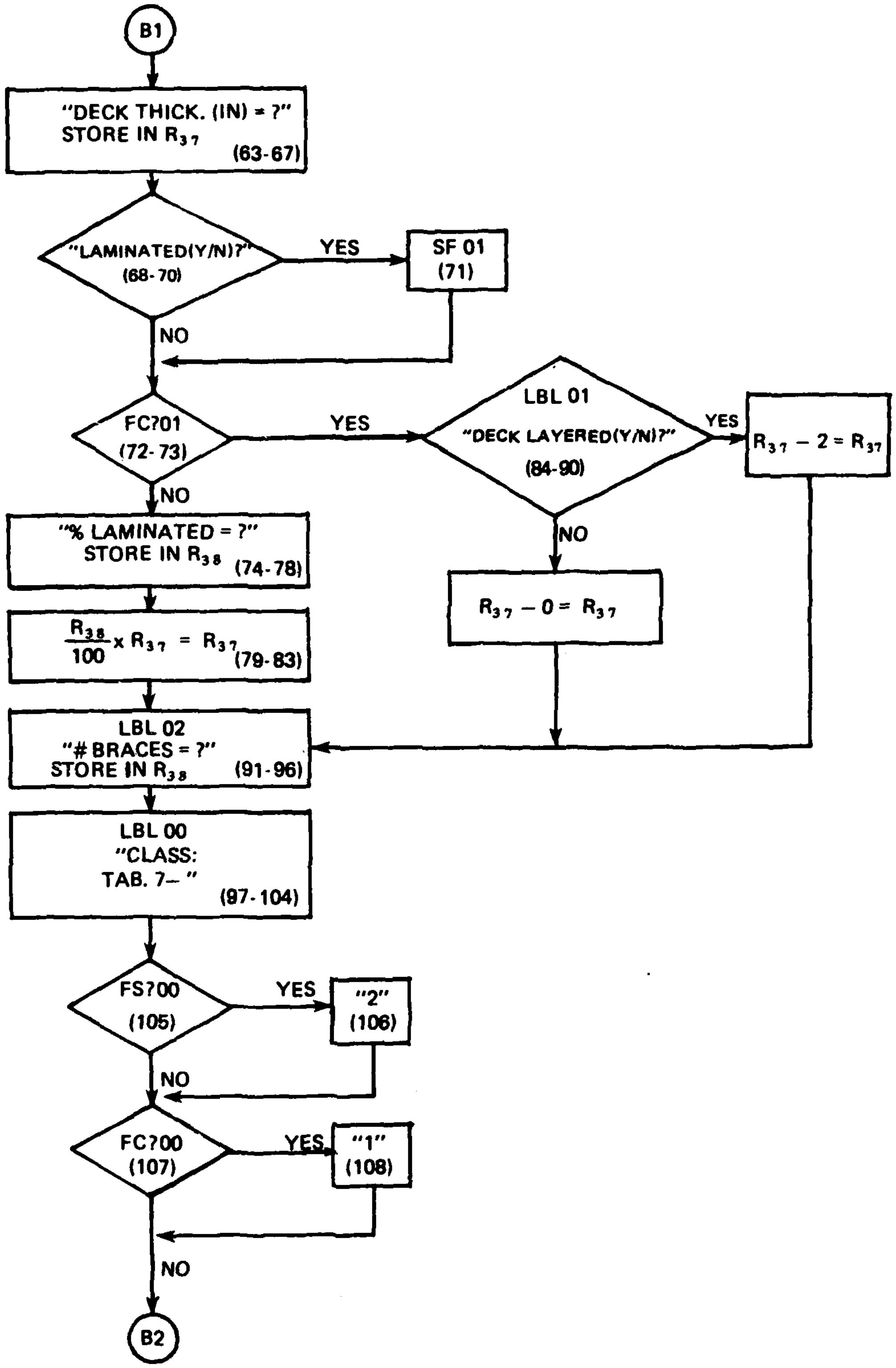


Figure A4. (Cont'd).

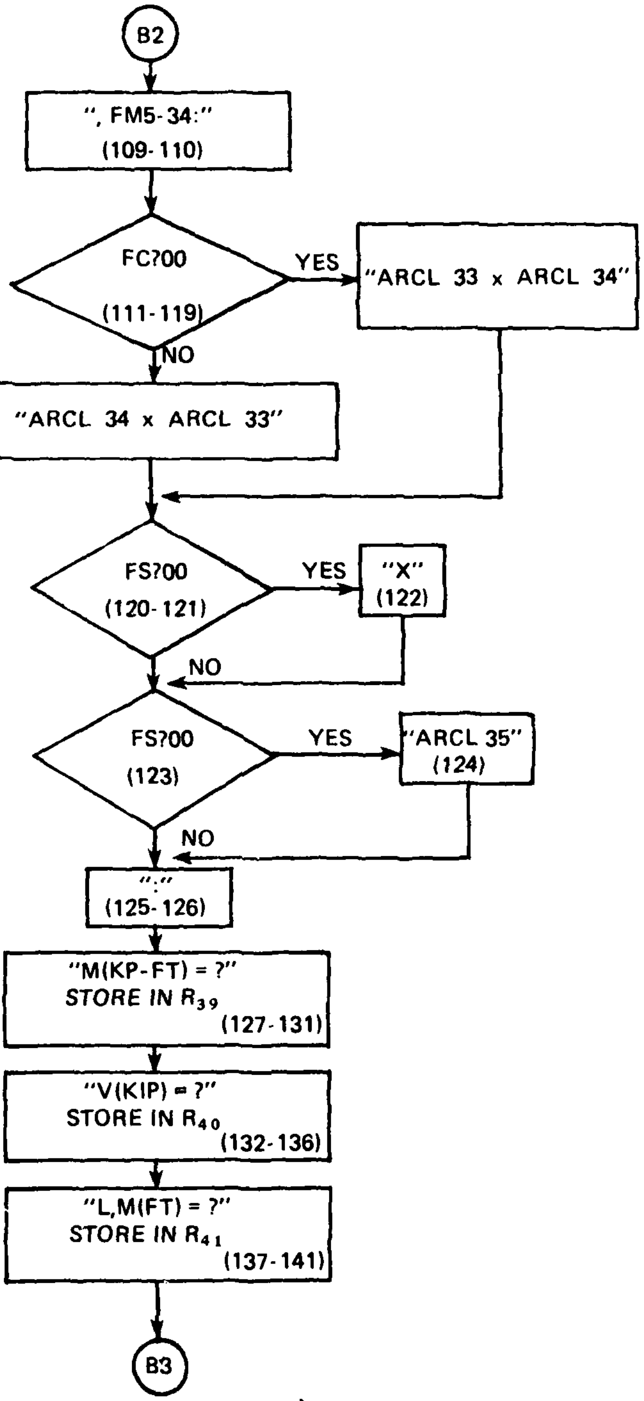


Figure A4. (Cont'd).

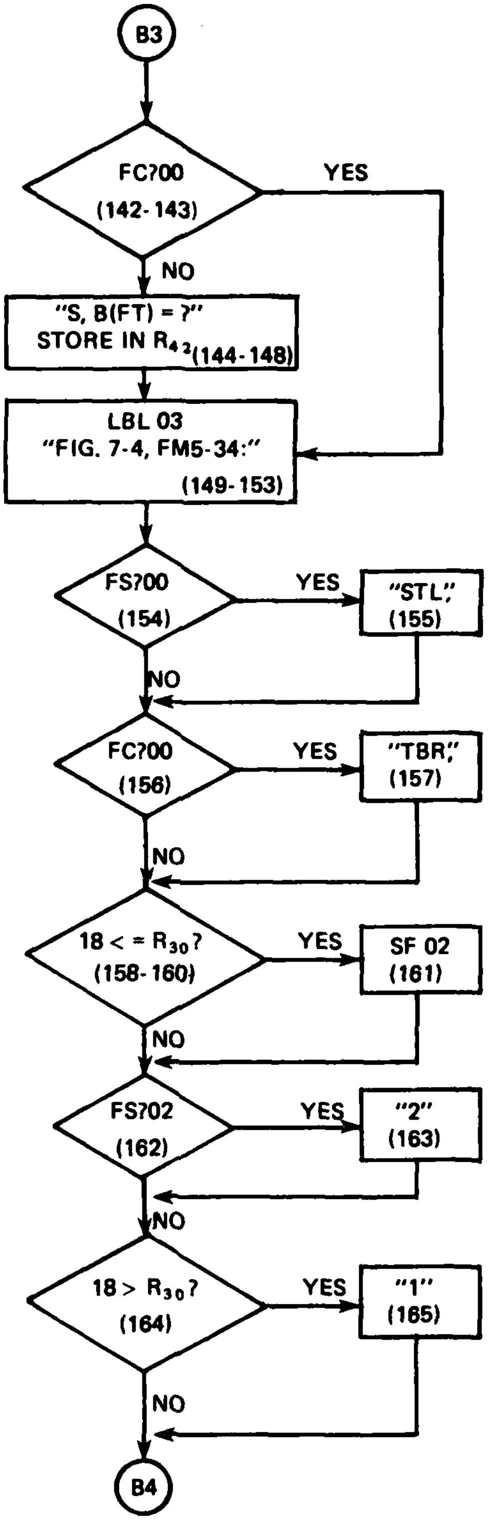


Figure A4. (Cont'd).

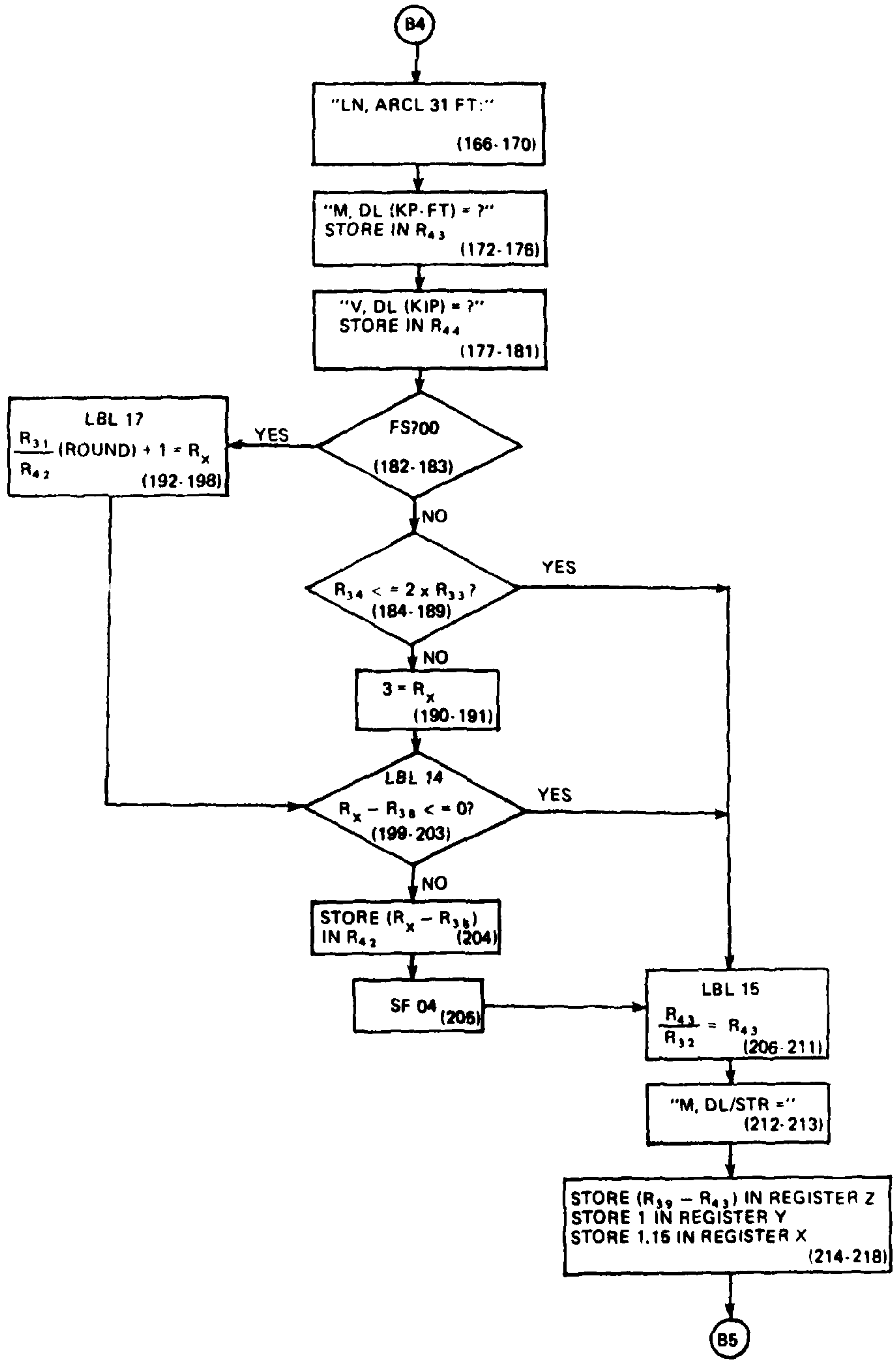


Figure A4. (Cont'd).

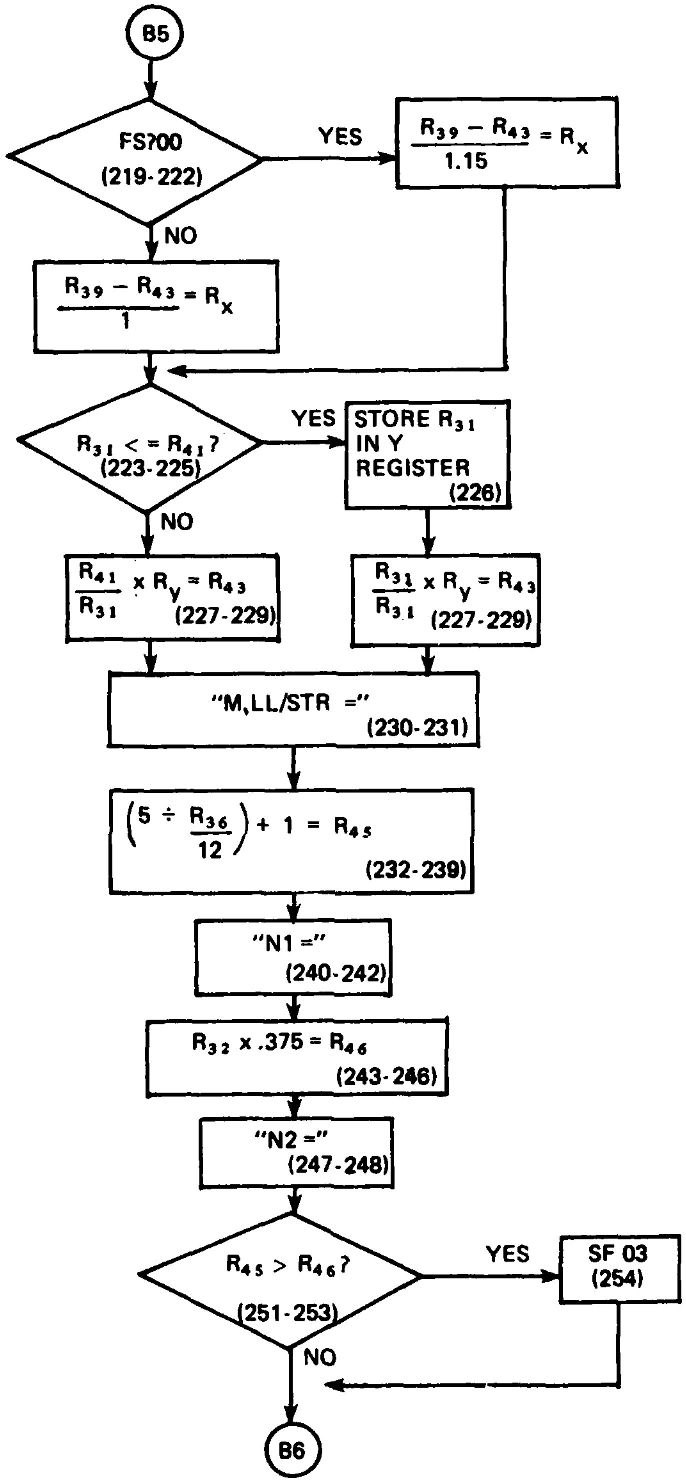


Figure A4. (Cont'd).

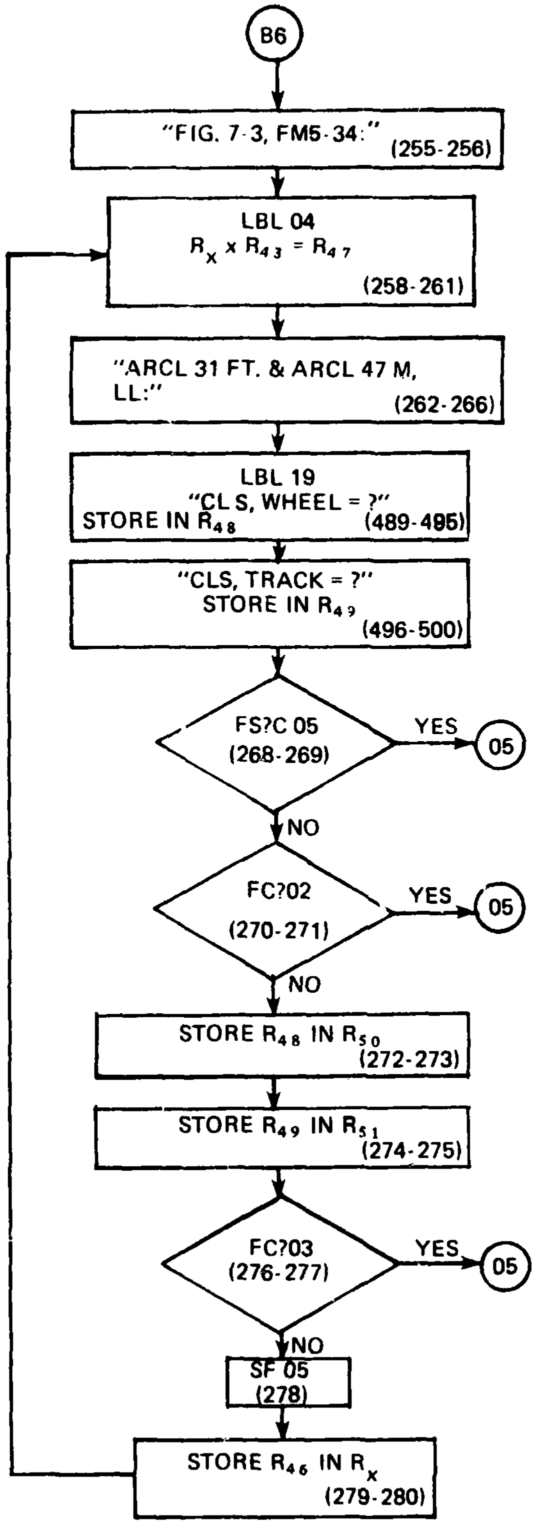


Figure A4. (Cont'd).

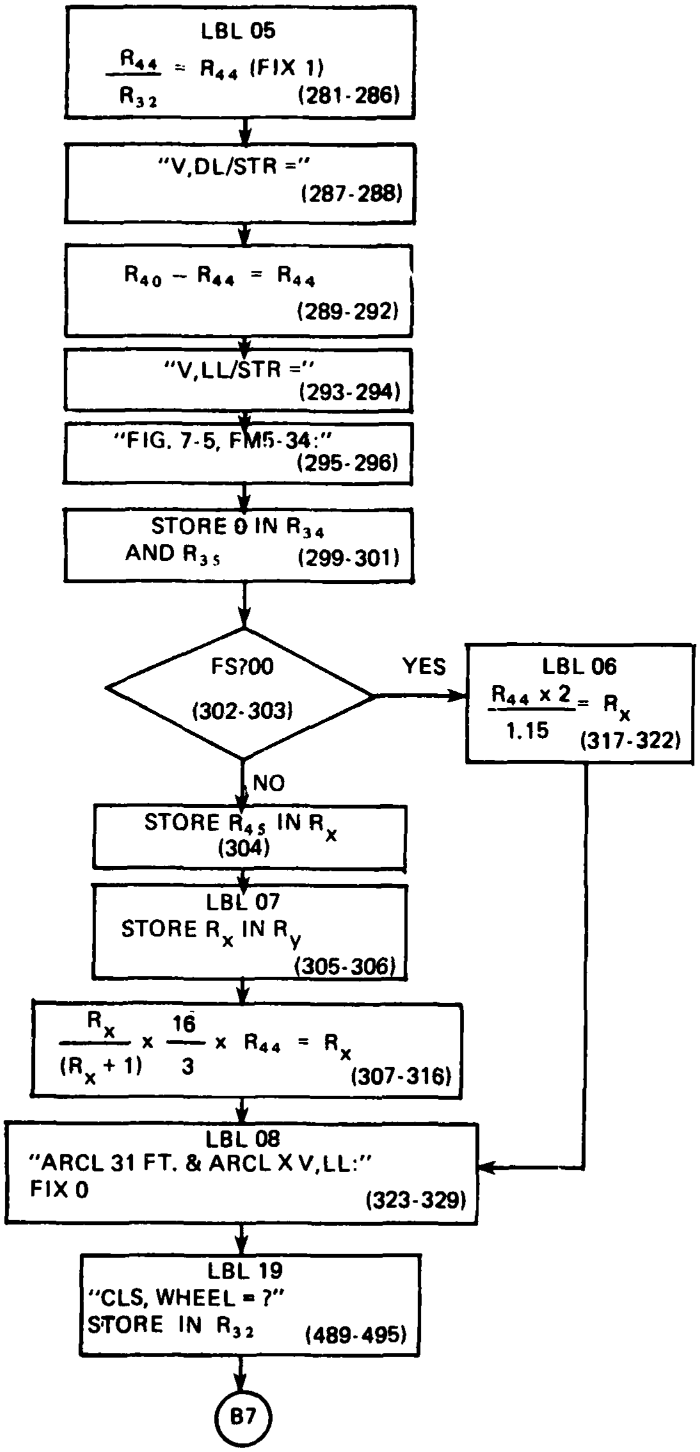


Figure A4. (Cont'd).

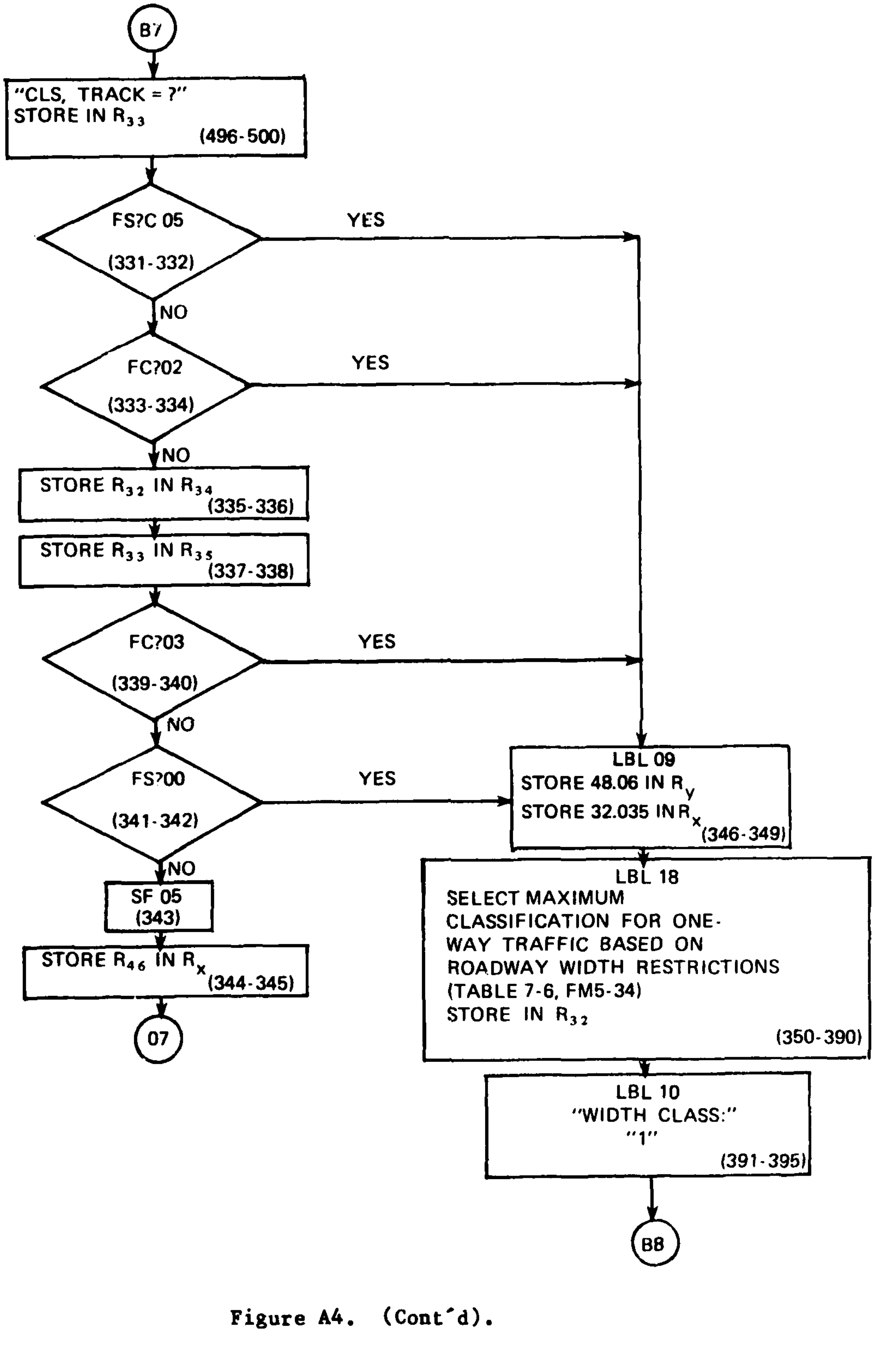


Figure A4. (Cont'd).

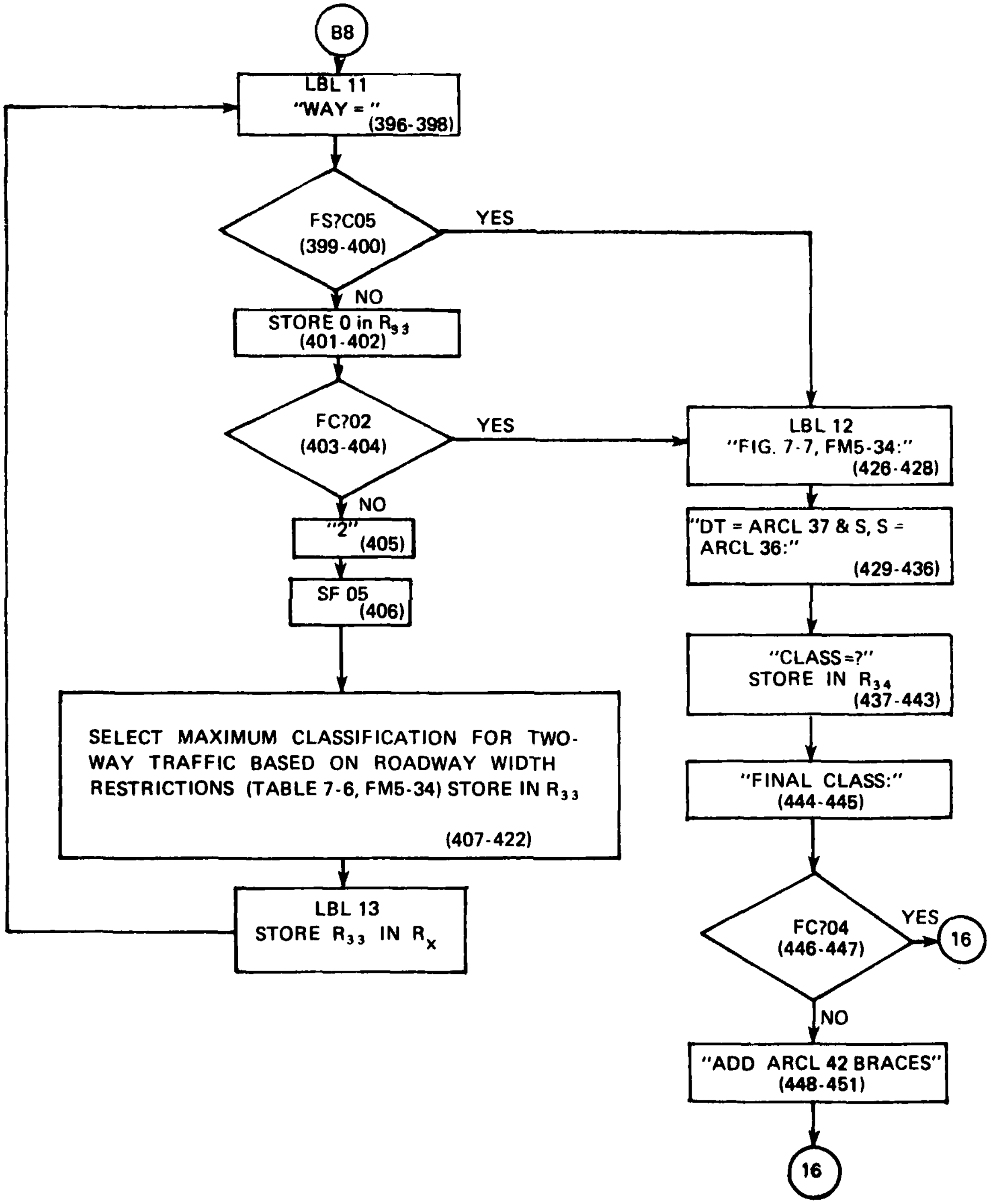


Figure A4 (Cont'd).

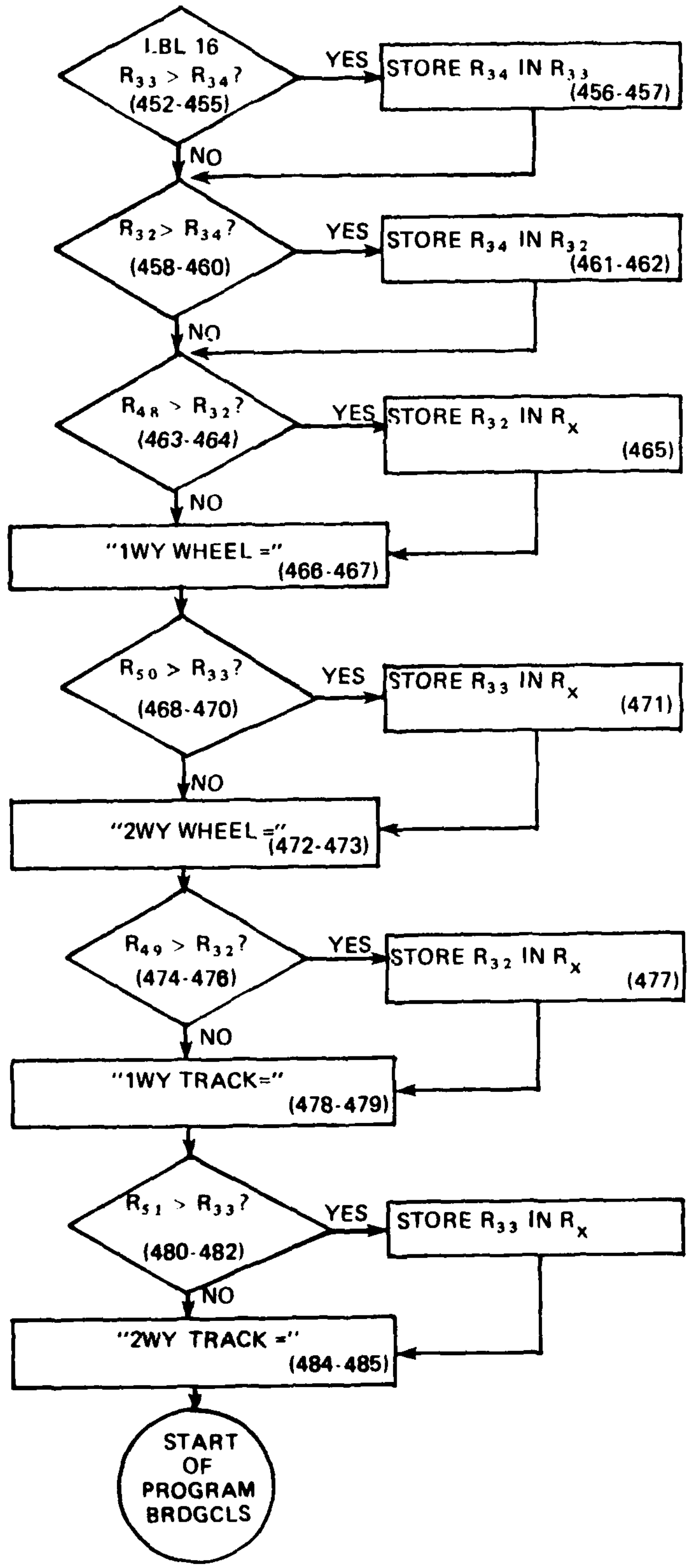


Figure A4. (Cont'd).

```
01♦LBL "BRDGCLS"
02 52
03 XEQ "*S"

04♦LBL 20
05 XEQ "*F"
06 CF 09
07 "BRIDGE CLASS"
08 XEQ "*D"
09 FIX 0
10 "RECON:"
11 XEQ "*D"
12 30
13 STO 24
14 "ROAD WIDTH(FT)"
15 50
16 ENTER↑
17 8
18 XEQ "*I"
19 "SPAN LENGTH(FT)"
20 200
21 ENTER↑
22 10
23 XEQ "*I"
24 "STRINGERS:"
25 XEQ "*D"
26 "STR, NUMBER"
27 25
82 ENTER↑
29 2
30 XEQ "*I"
31 "STEEL"
32 XEQ "*Y"
33 FS? 10
34 SF 00
35 "STR. WIDTH(IN)"
36 20
37 ENTER↑
38 4
39 XEQ "*I"
40 "STR. DEPTH(IN)"
41 60
42 ENTER↑
43 6
44 XEQ "*I"
45 "STR. FLANGE THI"
46 "⊢CK(IN)"
47 FIX 1
48 2
49 ENTER↑
50 .3
51 FS? 00
52 XEQ "*I"
53 FIX 0
54 FC? 00
55 ISG 24
56 STO X
57 "STR. SPACING(IN"
58 "⊢)"
59 100
60 ENTER↑
61 10
62 XEQ "*I"
63 "DECK THICK.(IN)"
64 12
65 ENTER↑
66 2
67 XEQ "*I"
68 "LAMINATED"
69 XEQ "*Y"
70 FS? 10
71 SF 01
72 FC? 01
73 GTO 01
74 "% LAM."
75 100
76 ENTER↑
77 0
78 XEQ "*I"
79 100
80 /
81 ST* 37
82 DSE 24
83 GTO 02

84♦LBL 01
85 "DECK LAYERED"
86 XEQ "*Y"
87 0
88 FS? 10
89 2
90 ST- 37

91♦LBL 02
92 "#BRACES"
93 20
94 ENTER↑
95 0
96 XEQ "*I"
97♦LBL 00
98 39.051
99 XEQ "*C"
100 "CLASS:"
101 XEQ "*D"
102 39
103 STO 24
104 "TAB.7-"
105 FS? 00
106 "⊢2"
107 FC? 00
108 "⊢1"
109 "⊢,FM5-34:"
110 XEQ "*D"
111 FIX 1
112 CLA
113 RCL 33
114 RCL 34
115 FC? 00
116 X<>Y
117 ARCL X
118 "⊢X"
119 ARCL Y
120 FIX 2
121 FS? 00
122 "⊢X"
123 FS? 00
124 ARCL 35
125 "⊢:"
126 XEQ "*D"
127 "M(KP-FT)"
128 3100
129 ENTER↑
130 3
131XEQ "*I"
132 "V(KIP)"
133 600
134 ENTER↑
135 3
136 XEQ "*I"
137 "L,M(FT)"
138 135
139 ENTER↑
140 9
141 XEQ "*I"
142 FC? 00
143 GTO 03
144 "S,B(FT)"
145 26
```

Figure A5. "BRDGCLS" program listing.

```
146 ENTER↑
147 6
148 XEQ "*I"

149♦LBL 03
150 43
151 STO 24
152 "FIG.7-4,FM5-34:"
153 XEQ "*D"
154 FS? 00
155 "STL,"
156 FC? 00
157 "TBR,"
158 RCL 30
159 18
160 X<=Y?
161 SF 02
162 FS? 02
163 "⊢2"
164 X>Y?
165 "⊢1"
166 "⊢ LN,"
167 FIX 0
168 ARCL 31
169 "⊢FT:"
170 XEQ "*D"
171 FIX 2
172 "M,DL(KP-FT)"
173 1200
174 ENTER↑
175 3
176 XEQ "*I"
177 "V,DL(KIP)"
178 65
179 ENTER↑
180 1
181 XEQ "*I"
182 FS? 00
183 GTO 17
184 RCL 33
185 2
186 *
187 RCL 34
188 X<=Y?
189 GTO 15
190 3
191 GTO 14

192♦LBL 17
193 RCL 31
194 RCL 42
195 /
196 XEQ "*R"
197 1
198 +

199♦LBL 14
200 RCL 38
201 -
202 X<=0?
203 GTO 15
204 STO 42
205 SF 04

206♦LBL 15
207 FIX 1
208 RCL 43
209 RCL 32
210 /
211 STO 43
212 "M,DL/STR"
213 XEQ "*O"
214 RCL 39
215 RCL 43
216 -
217 1
218 1.15
219 FS? 00
220 X<>Y
221 RDN
222 /
223 RCL 41
224 RCL 31
225 X<=Y?
226 STO Y
227 /
228 *
229 STO 43
230 "M,LL/STR"
231 XEQ "*O"
232 5
233 RCL 36
234 12
235 /
236 /
237 1
238 +
239 STO 45
240 FIX 2
241 "N1"
242 XEQ "*O"
243 RCL 32
244 .375
245 *
246 STO 46
247 "N2"
248 XEQ "*O"
249 48
250 STO 24
251 RCL 46
252 RCL 45
253 X>Y?
254 SF 03
255 "FIG.7-3,FM5-34:"
256 XEQ "*D"
257 FIX 0

258♦LBL 04
259 RCL 43
260 *
261 STO 47
262 CLA
263 ARCL 31
264 "⊢FT.&"
265 ARCL 47
266 "⊢M,LL:"
267 XEQ 19
268 FS?C 05
269 GTO 05
270 FC? 02
271 GTO 05
272 RCL 48
273 STO 50
274 RCL 49
275 STO 51
276 FC? 03
277 GTO 05
278 SF 05
279 RCL 46
280 GTO 04

281♦LBL 05
282 RCL 44
283 RCL 32
284 /
285 FIX 1
286 STO 44
```

Figure A5. (Cont'd).

```
287 "V,DL/STR"
288 XEQ "*O"
289 RCL 40
290 RCL 44
291 -
292 STO 44
293 "V,LL/STR"
294 XEQ "*O"
295 "FIG.7-5,FM5-34:"
296 XEQ "*D"
297 32
298 STO 24
299 0
300 STO 34
301 STO 35
302 FS? 00
303 GTO 06
304 RCL 45

305♦LBL 07
306 STO Y
307 1
308 +
309 /
310 16
311 3
312 /
313 *
314 RCL 44
315 *
316 GTO 08

317♦LBL 06
318 2
319 RCL 44
320 *
321 1.15
322 /

323♦LBL 08
324 FIX 0
325 CLA
326 ARCL 31
327 "⊢FT.&"
328 ARCL X
329 "⊢V,LL:"
330 XEQ 19
331 FS?C 05
332 GTO 09
333 FC? 02
334 GTO 09
335 RCL 32
336 STO 34
337 RCL 33
338 STO 35
339 FC? 03
340 GTO 09
341 FS? 00
342 GTO 09
343 SF 05
344 RCL 46
345 GTO 07

346♦LBL 09
347 48.06
348 ENTER↑
349 32.035

350♦LBL 18
351 RCL IND Y
352 RCL IND Y
353 X>Y?
354 X<>Y
355 STO IND T
356 RDN
357 RDN
358 ISG Y
359 ISG X
360 GTO 18
361 0
362 STO 32
363 RCL 30
364 9
365 -
366 X<0?
367 GTO 10
368 12
369 STO 32
370 RDN
371 2
372 -
373 X<0?
374 GTO 10
375 30
376 STO 32
377 RDN
378 2.16
379 -
380 X<0?
381 GTO 10
382 60
383 STO 32
384 RDN
385 1.58
386 -
387 X<0?
388 GTO 10
389 100
390 STO 32

391♦LBL 10
392 RCL 32
393 "WIDTH CLS:"
394 XEQ "*D"
395 "1"
396♦LBL 11
397 "⊢ WAY"
398 XEQ "*O"
399 FS?C 05
400 GTO 12
401 0
402 STO 33
403 FC? 02
404 GTO 12
405 "2"
406 SF 05
407 30
408 STO 33
409 RCL 30
410 24
411 -
412 X<0?
413 GTO 13
414 60
415 STO 33
416 RDN
417 3
418 -
419 X<0?
420 GTO 13
421 100
422 STO 33

423♦LBL 13
424 RCL 33
425 GTO 11
```

Figure A5. (Cont'd).

```
426♦LBL 12
427 "FIG.7-7,FM5-34:"
428 XEQ "*D"
429 "DT="
430 FIX 1
431 ARCL 37
432 "⊢&S,S="
433 FIX 0
434 ARCL 36
435 "⊢:"
436 XEQ "*D"
437 "CLASS"
438 34
439 STO 24
440 150
441 ENTER↑
442 0
443 XEQ "*I"
444 "FINAL CLASS:"
445 XEQ "*D"
446 FC? 04
447 GTO 16
448 "ADD"
449 ARCL 42
450 "⊢ BRACES"
451 XEQ "*D"

452♦LBL 16
453 RCL 34
454 RCL 33
455 X>Y?
456 X<>Y
457 STO 33
458 RCL 34
459 RCL 32
460 X>Y?
461 X<>Y
462 STO 32
463 RCL 48
464 X>Y?
465 X<>Y
466 "1WY WHEEL"
467 XEQ "*O"
468 RCL 33
469 RCL 50
470 X>Y?
471 X<>Y
472 "2WY WHEEL"
473 XEQ "*O"
474 RCL 32
475 RCL 49
476 X>Y?
477 X<>Y
478 "1WY TRACK"
479 XEQ "*O"
480 RCL 33
481 RCL 51
482 X<Y?
483 X<>Y
484 "2WY TRACK"
485 XEQ "*O"
486 XEQ "*P"
487 STOP
488 GTO 20

489♦LBL 19
490 XEQ "*D"
491 "CLS, WHEEL"
492 150
493 ENTER↑
494 0
495 XEQ "*I"
496 "CLS, TRACK"
497 150
498 ENTER↑
499 0
500 GTO "*I"
501 "2/2/82"
502 .END.
```

Figure A5. (Cont'd).

Table A1

"BRDGCLS" Program Example -- Without Printer

Step	Press	Resulting Display
1	XEQ	
	ALPHA	
	BRDGCLS	
	ALPHA	BRIDGE CLASS
2	R/S	RECON:
3	R/S	ROAD WIDTH (FT) = ?
4	23 R/S	SPAN LENGTH (FT) = ?
5	17 R/S	STRINGERS:
6	R/S	STR, NUMBER = ?
7	9 R/S	STEEL (Y/N)?
8	N R/S	STR. WIDTH (IN) = ?
9	8 R/S	STR. DEPTH (IN) = ?
10	18 R/S	STR. SPACING (IN) = ?
11	35 R/S	DECK THICK. (IN) = ?
12	6 R/S	LAMINATED (Y/N)?
13	N R/S	DECK LAYERED (Y/N)?
14	Y R/S	# BRACES = ?
15	2 R/S	CLASS:
16	R/S	TAB. 7-1, FM 5-34:
17	R/S	8.0X18.0
18	R/S	M(KP-FT) = ?
19	86.40 R/S	V (KIP) = ?
20	14.40 R/S	L, M (FT) = ?
21	21.50 R/S	FIG. 7-4, FM 5-34:
22	R/S	TBR, 2 LN, 17. FT:
23	R/S	M, DL (KP-FT) = ?
24	37.36 R/S	V, DL (KIP) = ?
25	8.75 R/S	M, DL/STR = 4.2
26	R/S	M, LL/STR= 82.2
27	R/S	N1 = 2.71
28	R/S	N2 = 3.38
29	R/S	FIG. 7-3, FM 5-34:
30	R/S	17 FT. & 223. M, LL:
31	R/S	CLS, WHEEL = ?
32	60 R/S	CLS, TRACK = ?
33	40 R/S	V, DL/STR = 1.0
34	R/S	V, LL/STR = 13.4
35	R/S	FIG. 7-5, FM 5-34:
36	R/S	17 FT. & 52. V, LL:
37	R/S	CLS, WHEEL = ?
38	50 R/S	CLS, TRACK = ?
39	40 R/S	WIDTH CLS:
40	R/S	1 WAY = 100
41	R/S	2 WAY = 30
42	R/S	FIG. 7-7, FM 5-34:
43	R/S	DT = 4.0 & S,S = 35.:
44	R/S	CLASS = ?

Table A1 (Cont'd)

Step	Press	Resulting Display
45	12 R/S	FINAL CLASS:
46	R/S	ADD 1. BRACES
47	R/S	1 WY WHEEL = 12.
48	R/S	2 WY WHEEL = 12.
49	R/S	1 WY TRACK = 12.
50	R/S	2 WY TRACK = 12.
51	R/S	END PROGRAM

Table A2

"BRDGCLS" Program -- Special Register Uses

Register Number		Register Contents
30		Roadway width in feet
31		Span length in feet
32		Number of stringers
	(or)	One-way class, wheel, V
	(or)	1-way width class
33		Stringer width in inches
	(or)	One-way class, track, V
	(or)	Two-way width class
34		Stringer depth in inches
	(or)	Two-way class, wheel, V
	(or)	Deck class
35		Stringer flange thickness in inches*
	(or)	Two-way class, track, V
36		Stringer spacing in inches
37		Deck thickness in inches
	(or)	Effective deck thickness in inches
38		Percent laminated
	(or)	Number of braces
39		Moment capacity in kip-feet
40		Shear capacity in kips
41		Maximum span length in feet
42		Maximum bracing spacing in feet*
	(or)	Additional braces required
43		Dead load moment in kip-feet
	(or)	Dead load moment/stringer in kip-feet
	(or)	Live load moment/stringer in kip-feet
44		Dead load shear in kip-feet
	(or)	Dead load shear/stringer in kip-feet
	(or)	Live load shear/stringer in kip-feet
45		Effective number of stringers per lane
46		Effective number of stringers per lane for a two-lane bridge
47		Live load moment/lane in kip-feet/lane
48		One-way class, wheel, M
49		One-way class, track, M
50		Two-way class, track, M
51		Two-way class, track, M

*(Only if steel stringers are used)

Table A3

"BRDGCLS" Program -- Functions, by Label

Label	Purpose
BRDGCLS	Marks beginning of program; checks size allocation
00	Clears registers; displays classification of stringer and table in which the stringer properties are found; inputs moment capacity, shear capacity, maximum span length and maximum bracing spacing (for steel stringers)
01	Determines if deck is layered
02	Inputs number of braces needed
03	Displays "Fig. 7-4, FM 5-34:", type of stringer, number of lanes and span length; outputs dead load moment and dead load shear
04	Displays length of span and live load bending moment
05	Computes/outputs dead load shear per stringer and live load shear per stringer; displays "Fig. 7-5, FM 5-34"
06	Determines live load shear per lane for steel stringers
07	Determines live load shear per lane for timber stringers
08	Displays length of span and live load shear per lane
09	Intermediate calculation
10	Displays "WIDTH CLS:"
11	Computes/outputs maximum classification for two-way traffic based on roadway width
12	Displays "Fig. 7-7, FM 5-34:", decking thickness, stringer spacing, "FINAL CLASS:", and "ADD BRACES:"; inputs decking class
13	Intermediate calculation
14	Decides if additional braces are needed
15	Computes/outputs dead load moment per stringer, live load moment per stringer, effective number of stringers per lane and effective number of stringers per lane for a two-lane bridge; displays "Fig. 7-3, FM 5-34:"

Table A3 (Cont'd)

Label	Purpose
16	Computes/outputs final classifications for one-way wheeled, two-way wheeled, one-way tracked, and two-way tracked vehicles
17	Computes number of braces needed when steel stringers are used
18	Determines maximum classification for one-way traffic based on roadway width
19	Outputs wheel and track classification
20	Inputs road width, span length, stringer number, stringer width, stringer depth, stringer flange thickness, stringer spacing, deck thickness, and percent laminated

APPENDIX B:
"CPM" PROGRAM DETAILS

This appendix provides detailed information about the "CPM" program. Figure B1 shows the typical sequence of events and the options that a user encounters when executing this program. Table B1 is a sample problem showing the specific steps that must be followed when one uses the HP-41 calculator without a printer attached. In this example, the program determines the early start, early finish, late start, late finish and total float for a network consisting of five activities.

Duration and preceding activity numbers are input for each activity. Once this information is entered, numbers for the ending activities and starting activity are input. In steps 28 through 67 of Table B1, the user has to press the R/S key to restart the program after each message has been displayed. If a printer were attached and in the "NORM" printer mode, the program would advance automatically. The results output by the printer are shown in boxes, each containing the results for one activity (Figure B2). The printout can be cut apart to form a network. Preceding activities are listed in the upper left hand corner of the diagram so the user can place the activities in the correct order. A row of asterisks along the bottom of a box indicates that the activity lies on the critical path.

Abbreviations used in "CPM" are:

Symbol	Meaning
ACT	Activity
CPM	Critical path method
EF	Early finish
END	Ending
ES	Early start
LF	Late finish
LS	Late start
#	Number
PRED	Preceding
TF	Total float
(Y/N)	(Yes/No)

Five sets of operating limits for variable input are in the program:

Variable	Minimum	Maximum
Total number of activities	1	98
Activity identification numbers	1*	98
Duration	0	1000
Preceding activities identification numbers	1*	98
Starting activity number	1	98

*0 is used to exit from the variable.

The limits set for preceding activity identification numbers show that any activity between 0 and 98 can be entered. However, a single activity must have no more than five preceding and 25 ending activities; otherwise, the registers will be disrupted. "CPM" is not programmed to check these two limits; the user must observe them himself. Critical assumptions and formulas are as follows:

1. Early start computation: ES = ES (preceding activity) + Duration (preceding activity). Note: When more than one activity comes to an activity node, the largest value for ES is used.

2. Early finish computation: EF = ES + Duration.

3. Late finish computation: LF (preceding activity) = LF (current activity) - Duration. Note: Starting with the ending activities of the network, the largest EF is found for all activities. This number is set equal to the LF for each "end activity." Working backwards using the formula for LF, the other LFs are determined accordingly.

4. Late start computation: LS = LF-Duration.

5. Total float computation: TF = LS-ES.

"CPM" uses registers 30 through [(2XACT#) +43] to store the values described in Table B2. The program uses two general purpose flags. Flag 01 is set if there are no more ending activities, and flag 15 is set if the total float equals zero.

Table B3 describes the general function of each part of the program, by label. Figure B3 is a label wiring diagram showing how the different parts of the program relate to each other. A circular loop on the diagram indicates a return to the same label. A two-headed arrow pointing to and from a subroutine indicates that the program executes it as a local subroutine then returns to the main program. Global subroutines, used by all the major application programs on the MILENG1/UTIL module, are not shown on the wiring diagram, but are described separately in Appendix G.

Figure B4 is a detailed flowchart of the "CPM" program, and Figure B5 lists the program steps.

User executes "CPM"

↓

User enters total number of activities

↓

User enters activity number, duration, and preceding activities for each activity

↓

User enters ending activities

↓

User enters starting activity number

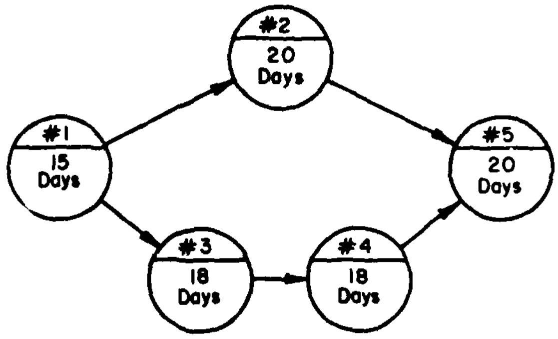
↓

Program computes and outputs early start, early finish, late start, late finish, total float, and critical path

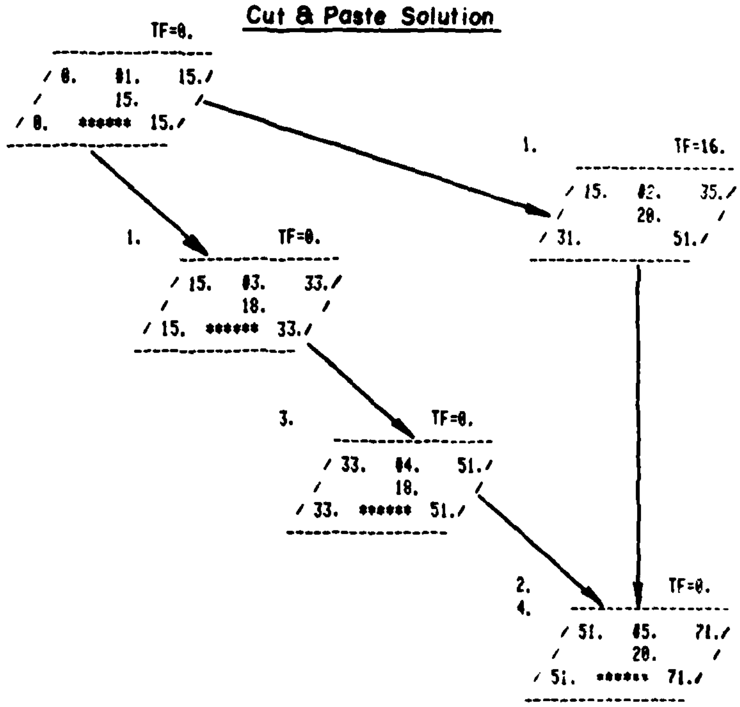
Figure B1. "CPM" program sequence of events.

```
                XEQ "CPM"
TOTAL #ACTIVITIES=?
               5.        RUN
ACTIVITY#=?
               1.        RUN
DURATION=?
              15.        RUN
PRED. ACT=?
               0.        RUN
ACTIVITY#=?
               2.        RUN
DURATION=?
              20.        RUN
PRED. ACT=?
               1.        RUN
PRED. ACT=?
               0.        RUN
ACTIVITY#=?
               3.        RUN
DURATION=?
              16.        RUN
PRED. ACT=?
               1.        RUN
PRED. ACT=?
               0.        RUN
ACTIVITY#=?
               4.        RUN
DURATION=?
              18.        RUN
PRED. ACT=?
               3.        RUN
PRED. ACT=?
               0.        RUN
ACTIVITY#=?
               5.        RUN
DURATION=?
              20.        RUN
PRED. ACT=?
               2.        RUN
PRED. ACT=?
               4.        RUN
PRED. ACT=?
               0.        RUN
ACTIVITY#=?
               0.        RUN
ANY CHANGES(Y/N)?
N                        RUN
 ENTER ENDING ACTIVITIES
END ACT.#=?
               5.        RUN
END ACT.#=?
               0.        RUN
START ACT.#=?
               1.        RUN
SEE KEY(Y/N)?
Y                        RUN
```

Simple Network



Cut & Paste Solution




```
   -----------------
  /ES  ACT#   EF/
 /   DURATION   /
/LS          LF/
-----------------
```

Figure B2. "CPM" program example -- with printer.

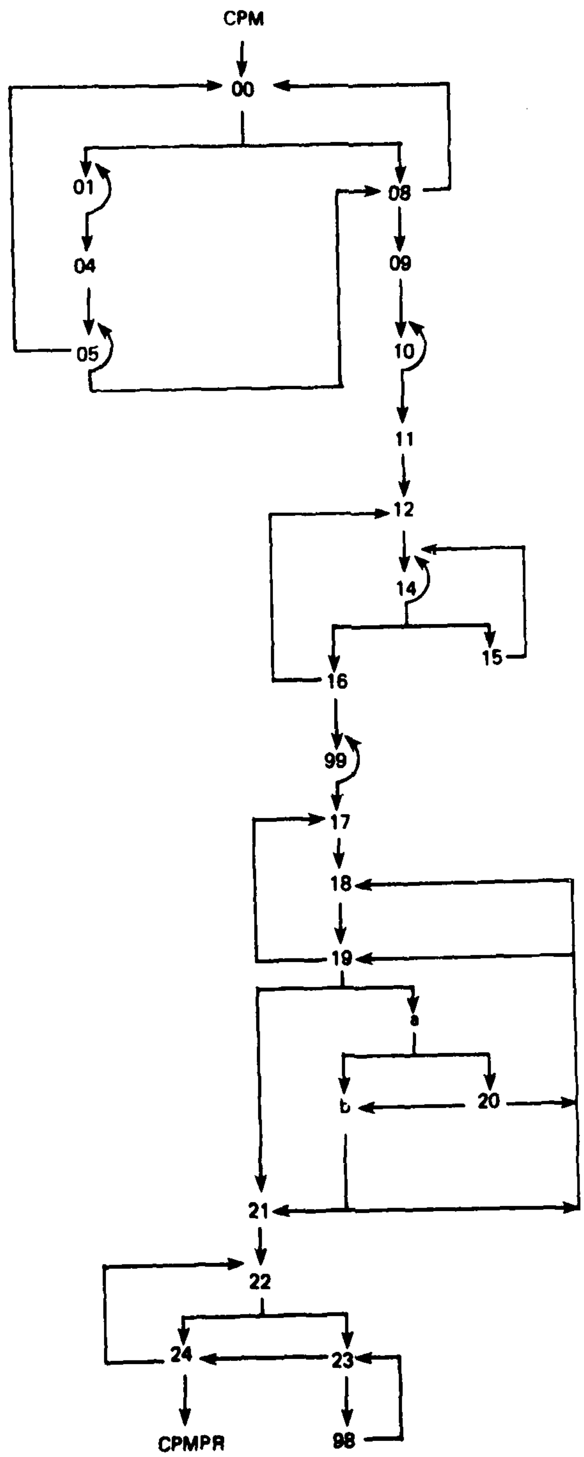


Figure B3. "CPM" program label wiring diagram.

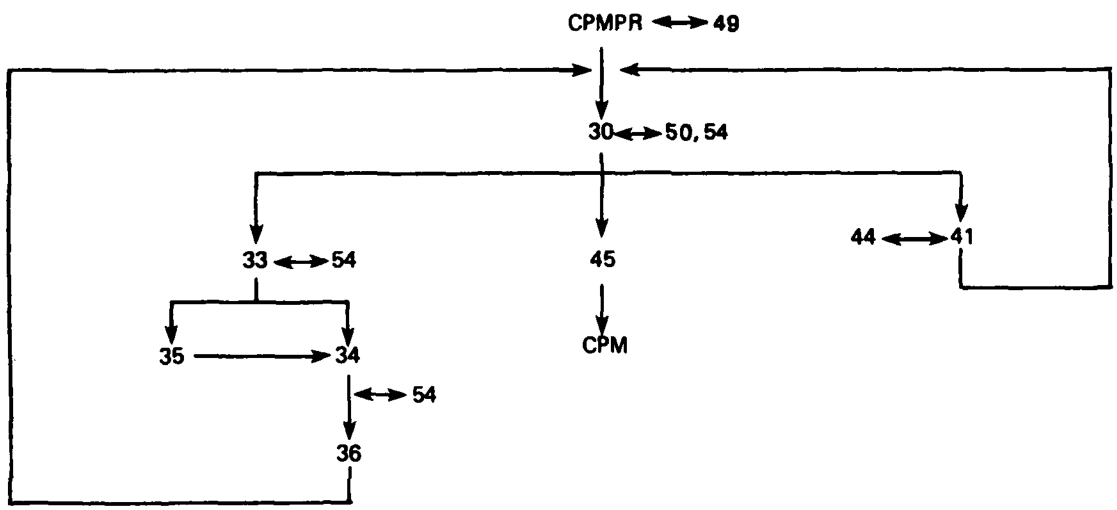


Figure B3. (Cont'd).

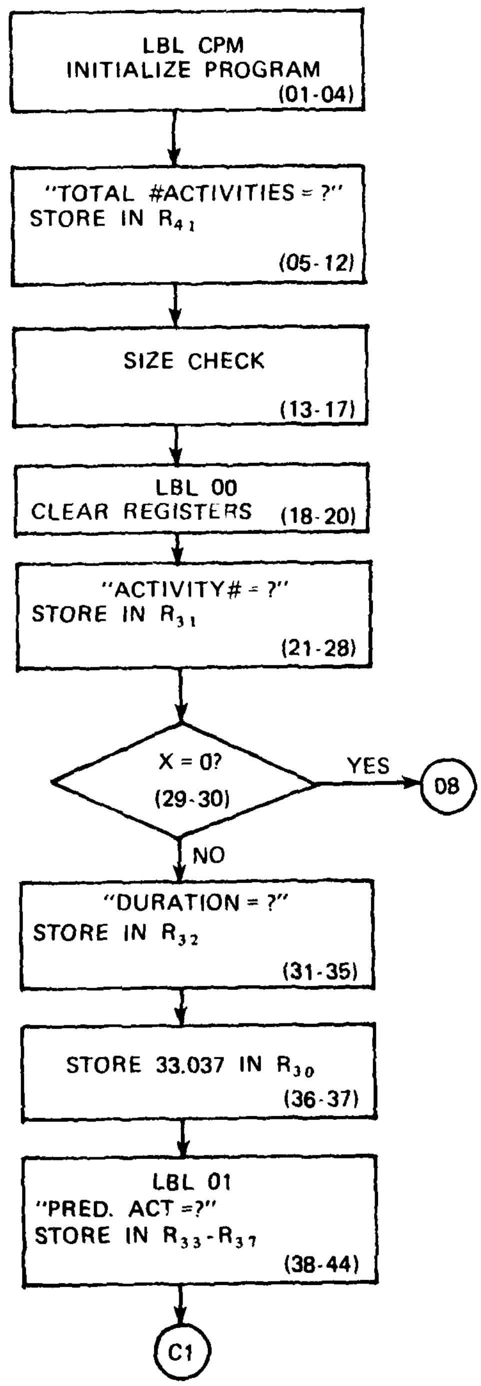


Figure B4. "CPM" program -- detailed flowchart.

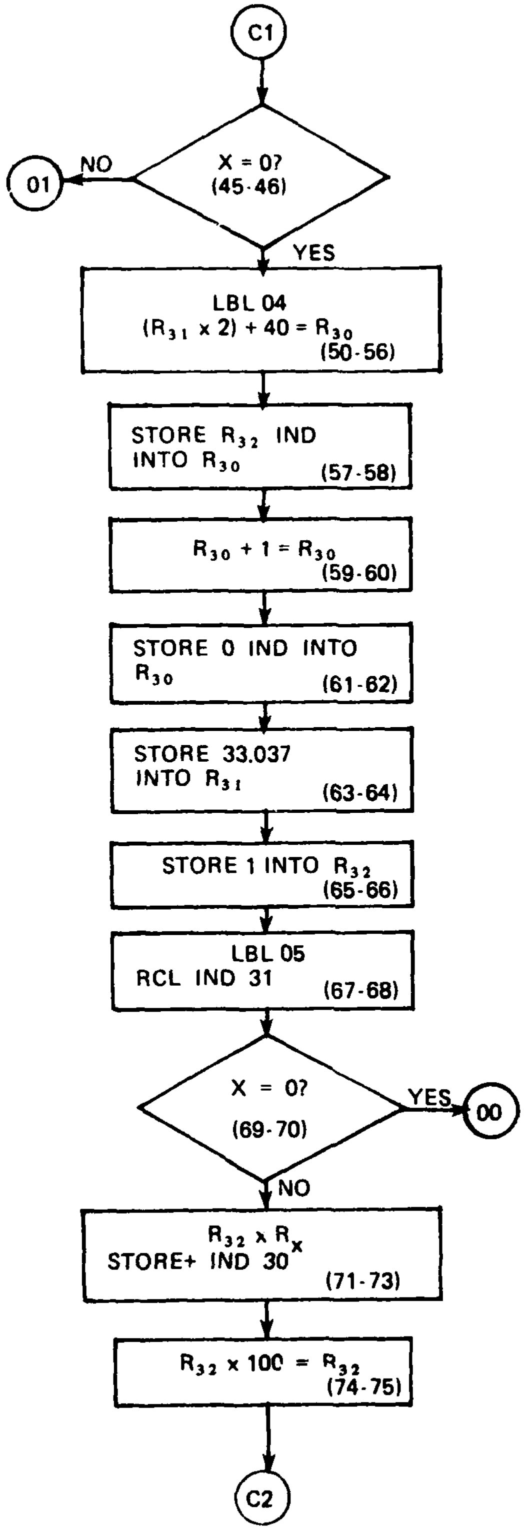


Figure B4. (Cont'd).

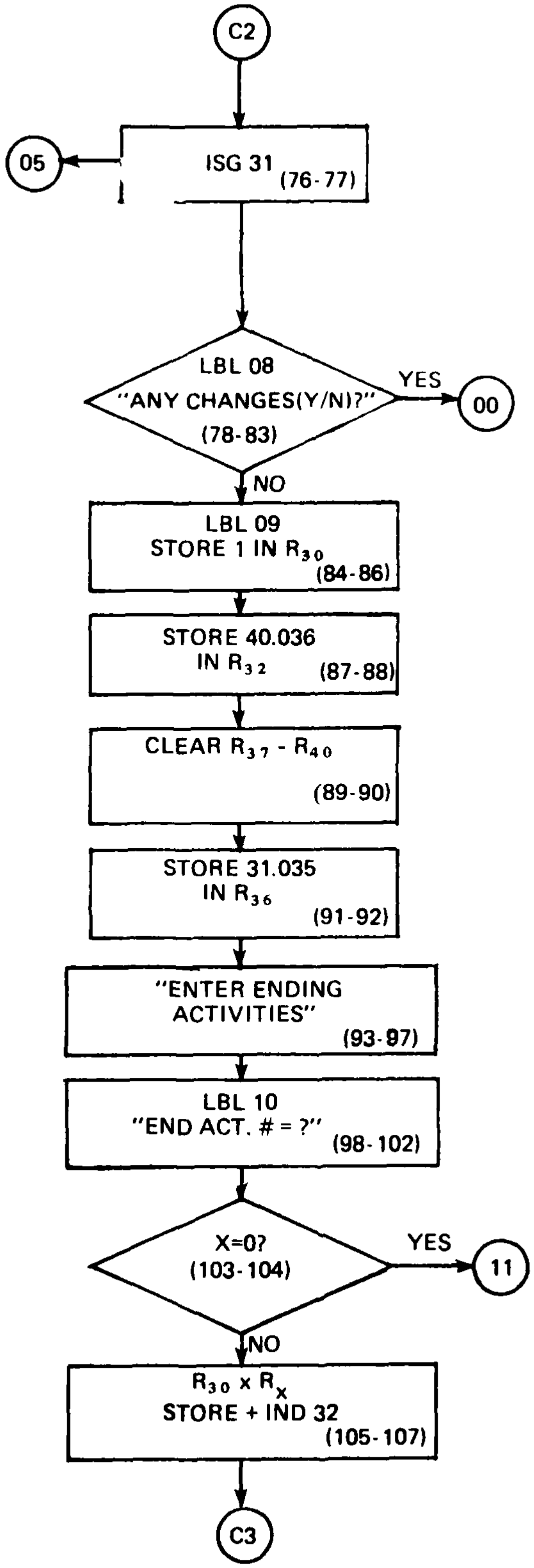


Figure B4. (Cont´d).

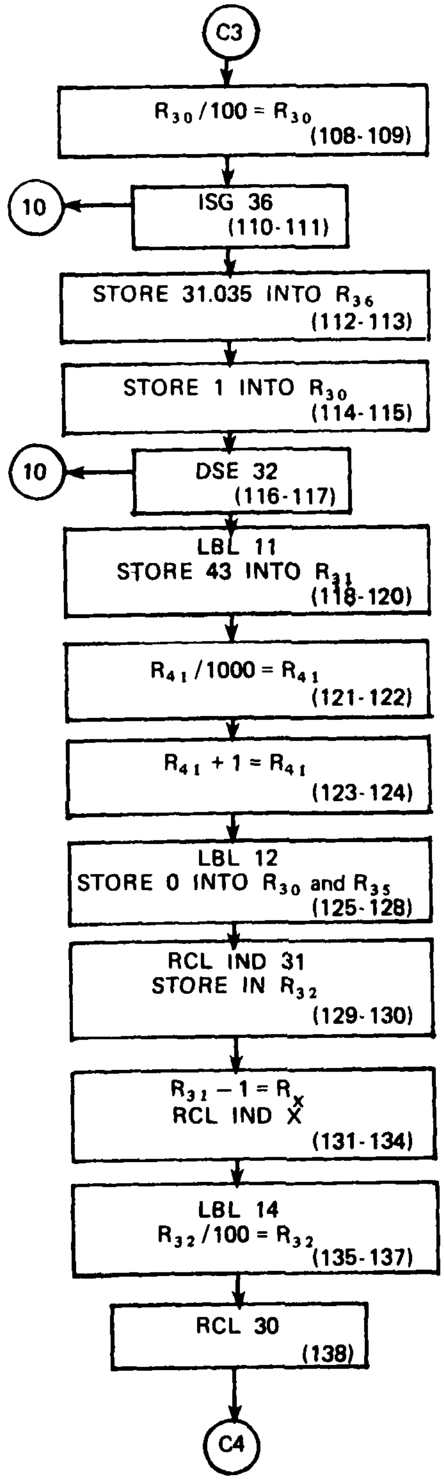


Figure B4. (Cont'd).

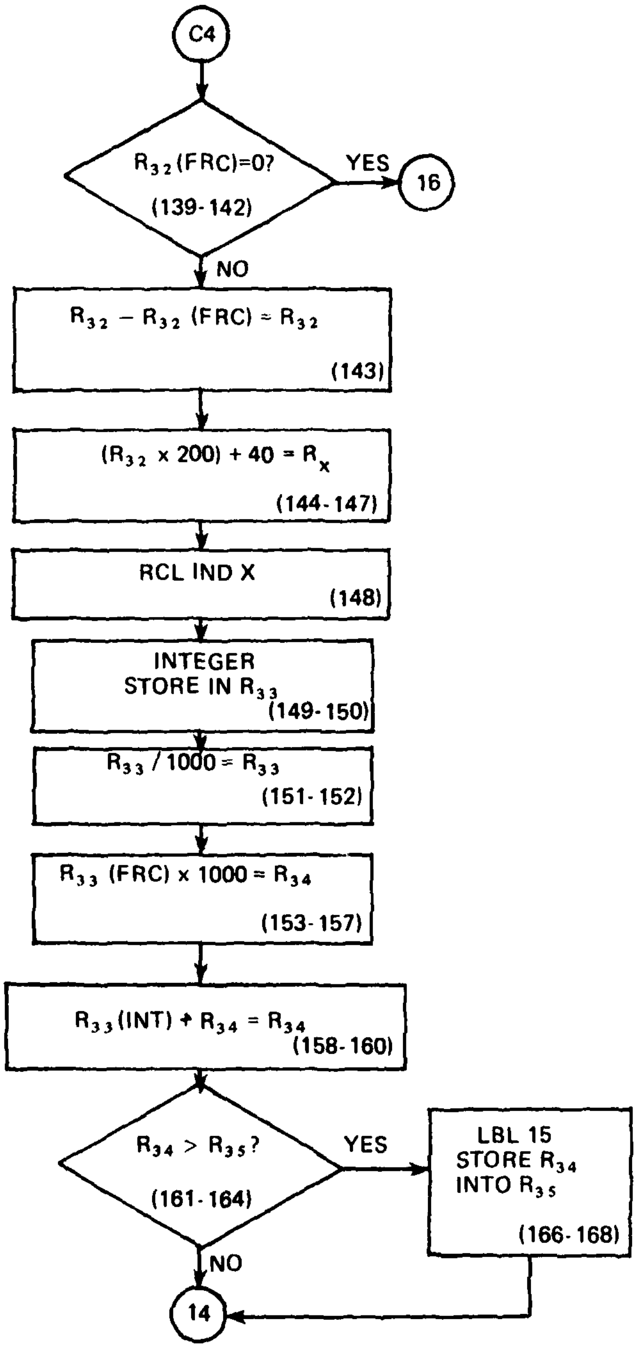


Figure B4. (Cont'd).

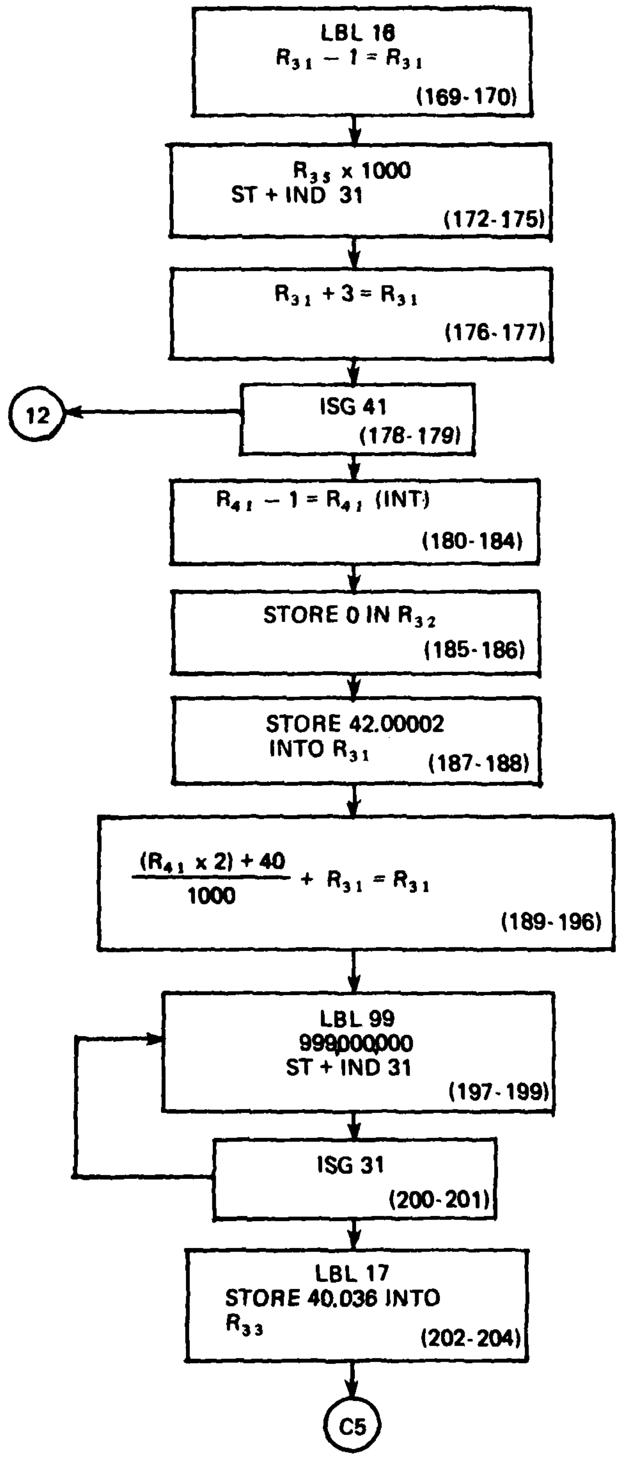


Figure B4. (Cont'd).

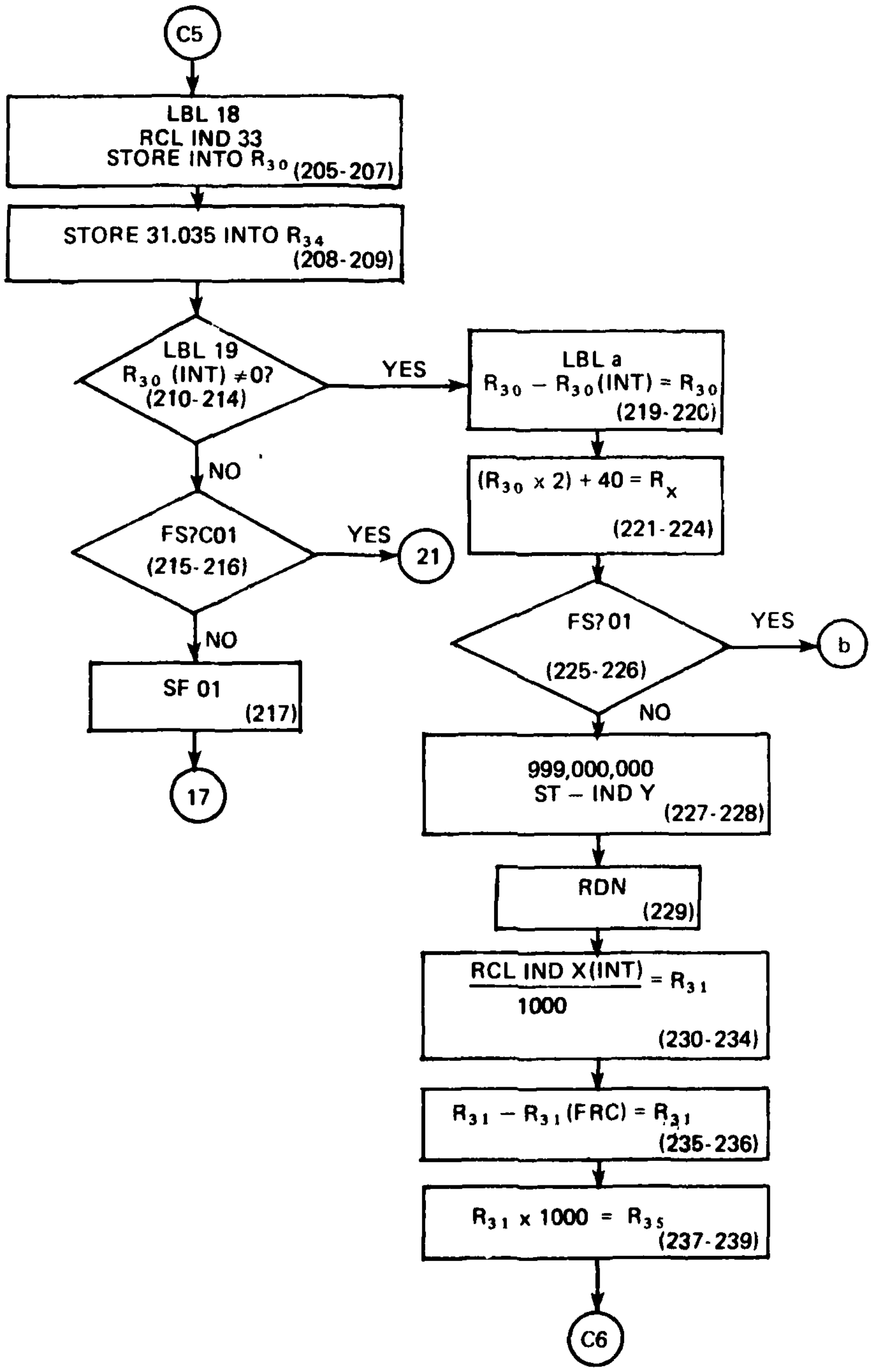


Figure B4. (Cont'd).

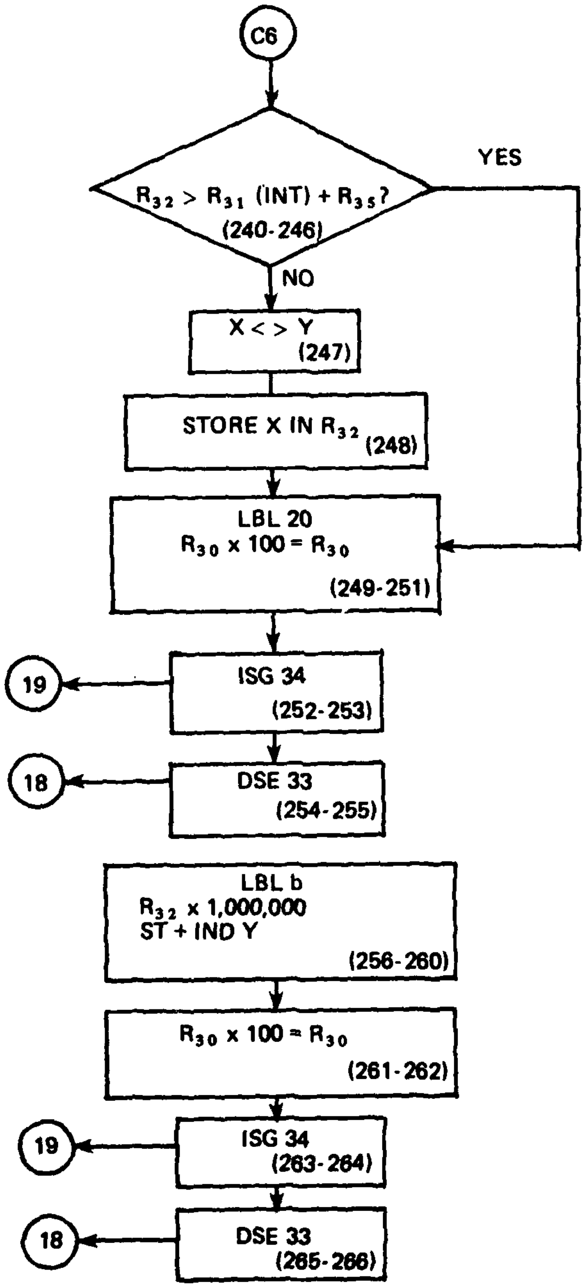


Figure B4. (Cont'd).

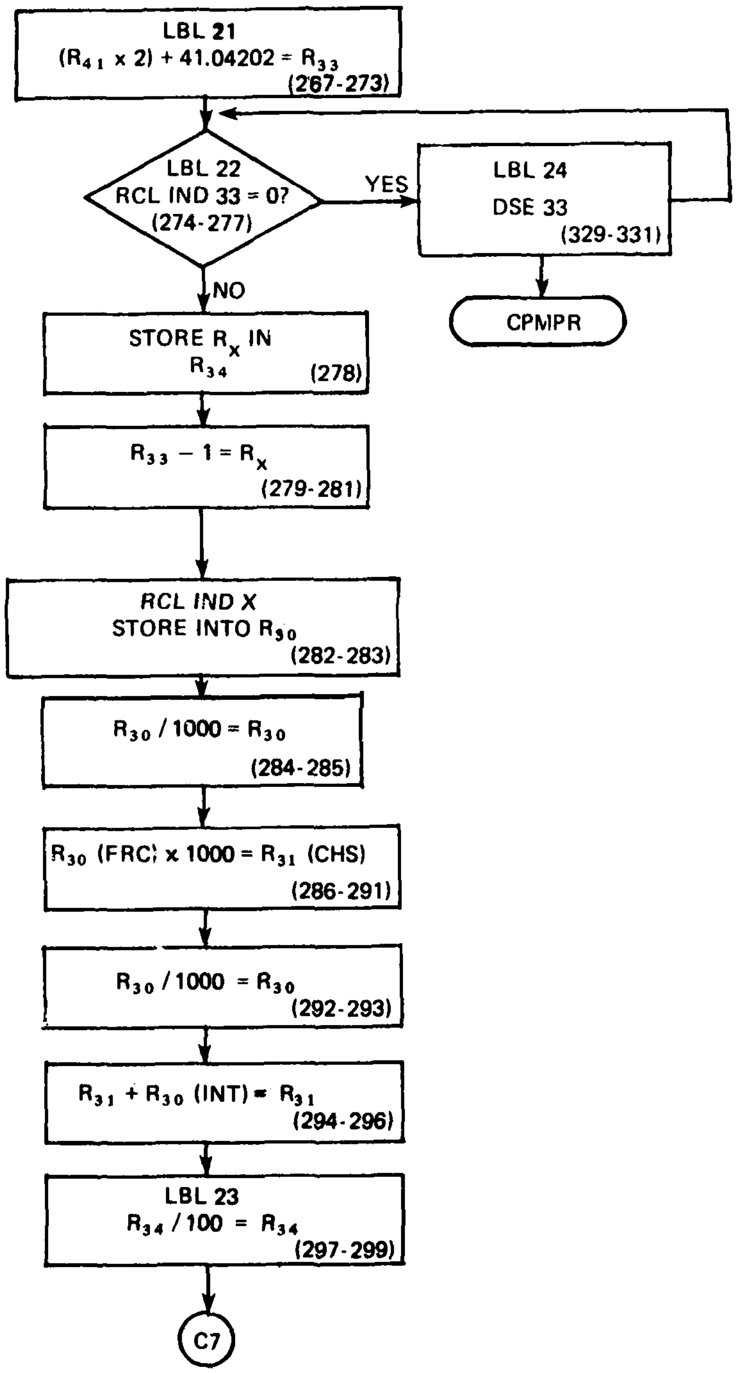


Figure B4. (Cont'd).

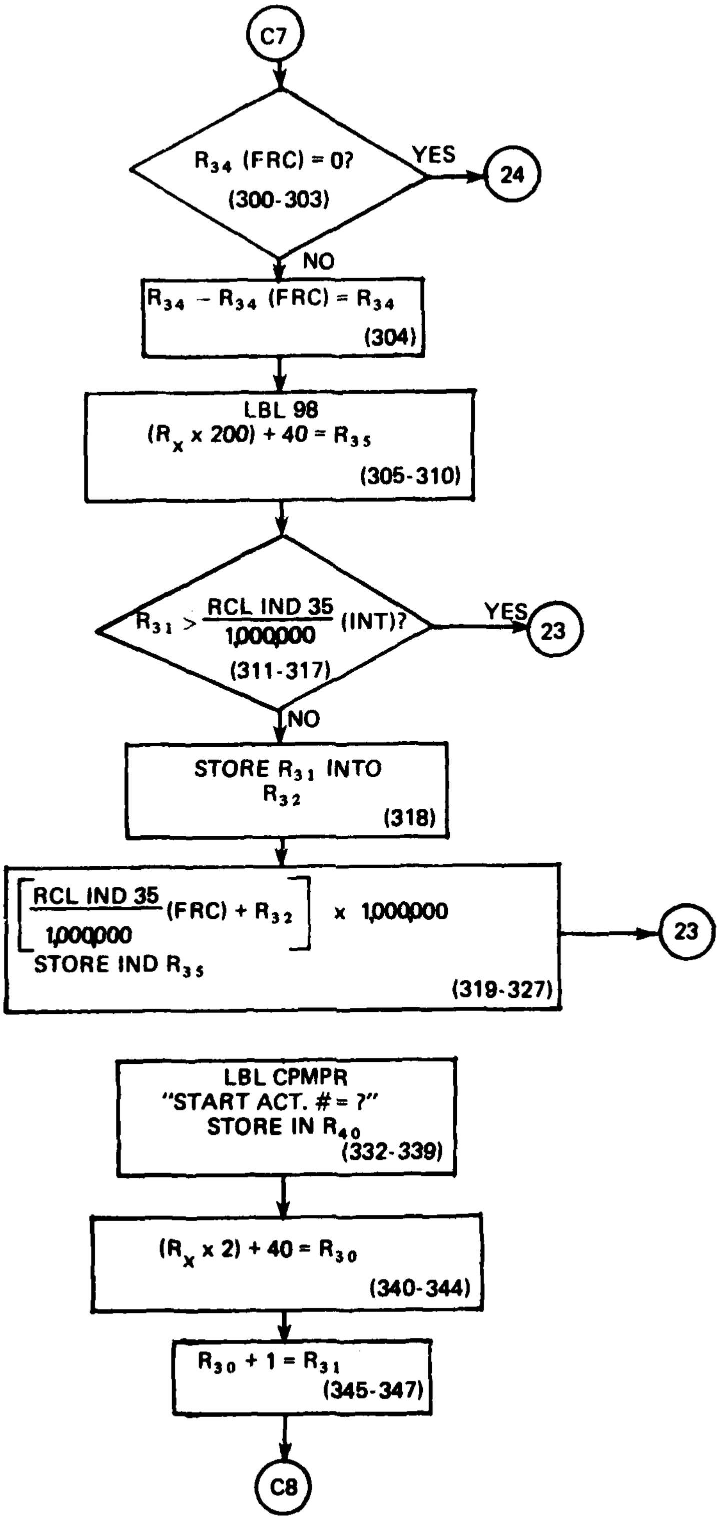


Figure B4. (Cont'd).

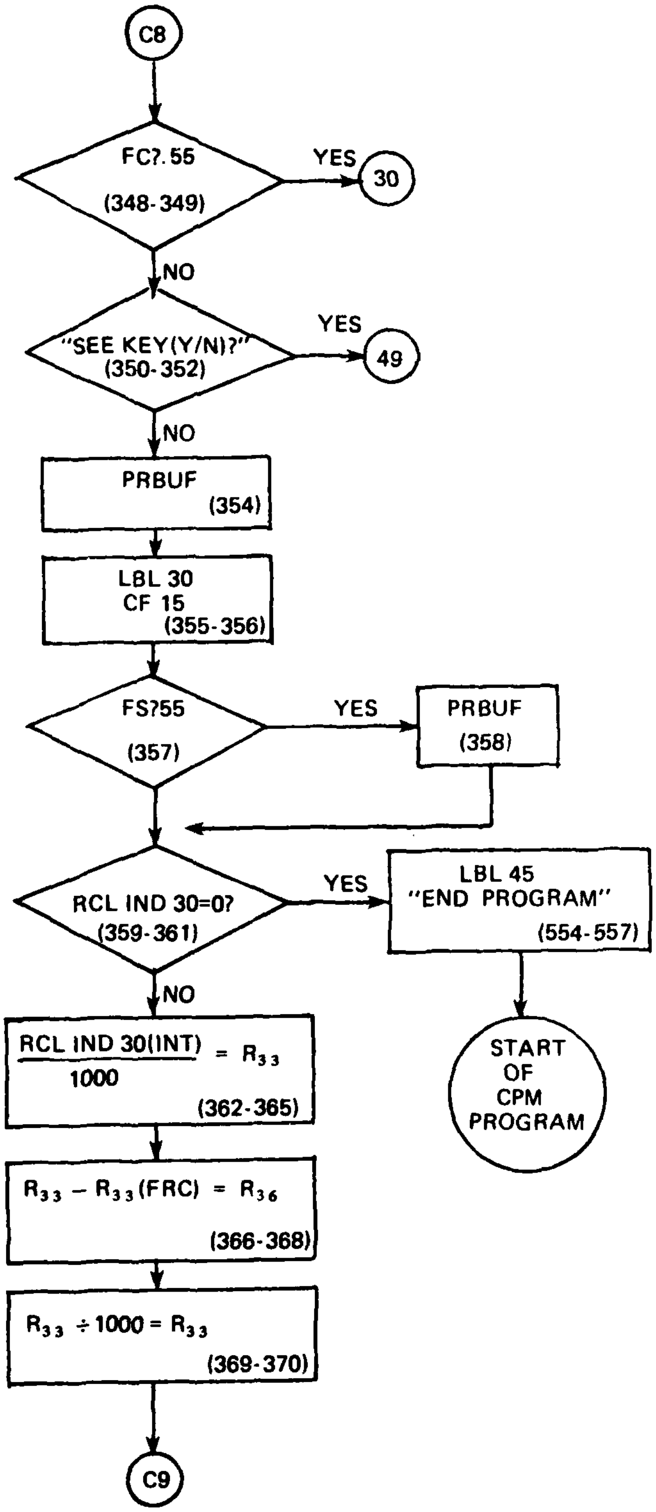


Figure B4. (Cont'd).

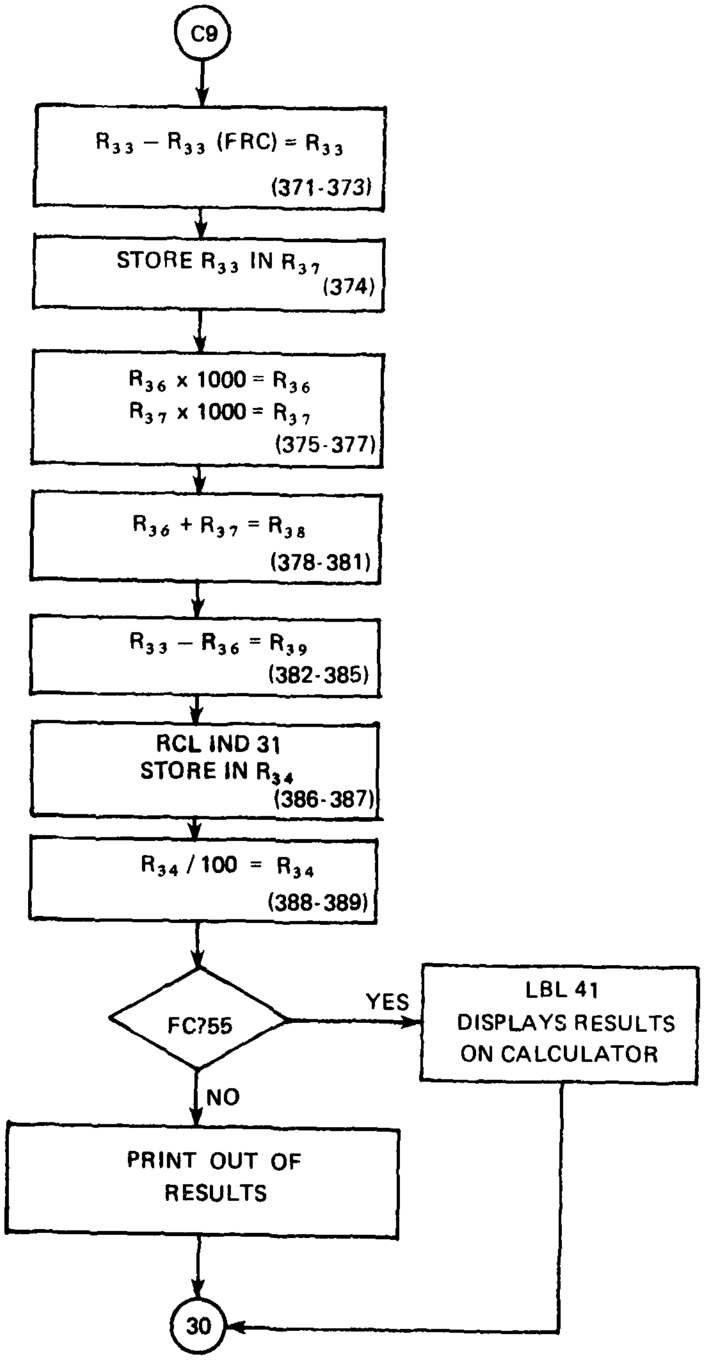


Figure B4. (Cont'd).

```
01♦LBL "CPM"
02 CLA
03 CLRG
04 CF 01
05 41
06 STO 24
07 "TOTAL #ACTIVITI"
08 "⊢ES"
09 98
10 ENTER↑
11 1
12 XEQ "*I"
13 2
14 *
15 43
16 +
17 XEQ "*S"

18♦LBL 00
19 33.040
20 XEQ "*C"
21 31
22 STO 24
23 0
24 "ACTIVITY#"
25 98
26 ENTER↑
27 0
28 XEQ "*I"
29 X=0?
30 GTO 08
31 "DURATION"
32 1000
33 ENTER↑
34 0
35 XEQ "*I"
36 33.037
37 STO 30

38♦LBL 01
39 0
40 "PRED. ACT"
41 98
42 ENTER↑
43 0
44 XEQ "*I"
45 X=0?
46 GTO 04
47 STO IND 30
48 ISG 30
49 GTO 01

50♦LBL 04
51 RCL 31
52 2
53 *
54 40
55 +
56 STO 30
57 RCL 32
58 STO IND 30
59 1
60 ST+ 30
61 0
62 STO IND 30
63 33.037
64 STO 31
65 1
66 STO 32

67♦LBL 05
68 RCL IND 31
69 X=0?
70 GTO 00
71 RCL 32
72 *
73 ST+ IND 30
74 100
75 ST* 32
76 ISG 31
77 GTO 05

78♦LBL 08
79 0
80 "ANY CHANGES"
81 XEQ "*Y"
82 FS? 10
83 GTO 00

84♦LBL 09
85 1
85 STO 30
87 40.036
88 STO 32
89 37.040
90 XEQ "*C"
91 31.035
92 STO 36
93 TONE 8
94 TONE 9
95 "ENTER ENDING A"
96 "⊢CTIVITIES"
97 AVIEW

98♦LBL 10
99 0
100 TONE 5
101 "END ACT.#=?"
102 PROMPT
103 X=0?
104 GTO 11
105 RCL 30
106 *
107 ST+ IND 32
108 100
109 ST/ 30
110 ISG 36
111 GTO 10
112 31.035
113 STO 36
114 1
115 STO 30
116 DSE 32
117 GTO 10

118♦LBL 11
119 43
120 STO 31
121 1000
122 ST/ 41
123 1
124 ST+ 41

125♦LBL 12
126 0
127 STO 35
128 STO 30
129 RCL IND 31
130 STO 32
131 RCL 31
132 1
133 -
134 RCL IND X

135♦LBL 14
136 100
137 ST/ 32
138 RCL 30
139 RCL 32
140 FRC
141 X=0?
142 GTO 16
143 ST- 32
144 200
145 *
146 40
147 +
```

Figure B5. "CPM" program listing.

```
148 RCL IND X
149 INT
150 STO 33
151 1000
152 ST/ 33
153 RCL 33
154 FRC
155 1000
156 *
157 STO 34
158 RCL 33
159 INT
160 ST+ 34
161 RCL 35
162 RCL 34
163 X>Y?
164 GTO 15
165 GTO 14

166♦LBL 15
167 STO 35
168 GTO 14

169♦LBL 16
170 1
171 ST- 31
172 RCL 35
173 1000
174 *
175 ST+ IND 31
176 3
177 ST+ 31
178 ISG 41
179 GTO 12
180 RCL 41
181 '
182 -
183 INT
184 STO 41
185 0
186 STO 32
187 42.00002
188 STO 31
189 RCL 41
190 2
191 *
192 40
193 +
194 1000
195 /
196 ST+ 31
197♦LBL 99
198 999000000
199 ST+ IND 31
200 ISG 31
201 GTO 99

202♦LBL 17
203 40.036
204 STO 33

205♦LBL 18
206 RCL IND 33
207 STO 30
208 31.035
209 STO 34

210♦LBL 19
211 RCL 30
212 INT
213 X≠0?
214 GTO a
215 FS?C 01
216 GTO 21
217 SF 01
218 GTO 17

219♦LBL a
220 ST- 30
221 2
222 *
223 40
224 +
225 FS? 01
226 GTO b
227 999000000
228 ST- IND Y
229 RDN
230 RCL IND X
231 INT
232 1000
233 /
234 STO 31
235 FRC
236 ST- 31
237 1000
238 *
239 STO 35
240 RCL 31
241 INT
242 RCL 35
243 +
244 RCL 32
245 X>Y?
246 GTO 20
247 X<>Y
248 STO 32

249♦LBL 20
250 100
251 ST* 30
252 ISG 34
253 GTO 19
254 DSE 33
255 GTO 18

256♦LBL b
257 RCL 32
258 1000000
259 *
260 ST+ IND Y
261 100
262 ST* 30
263 ISG 34
264 GTO 19
265 DSE 33
266 GTO 18

267♦LBL 21
268 RCL 41
269 2
270 *
271 41.04202
272 +
273 STO 33

274♦LBL 22
275 RCL IND 33
276 X=0?
277 GTO 24
278 STO 34
279 RCL 33
280 1
281 -
282 RCL IND X
283 STO 30
284 1000
285 ST/ 30
286 RCL 30
287 FRC
288 1000
289 *
290 CHS
```

Figure B5. (Cont'd).

```
291 STO 31
292 1000
293 ST/ 30
294 RCL 30
295 INT
296 ST+ 31

297♦LBL 23
298 100
299 ST/ 34
300 RCL 34
301 FRC
302 X=0?
303 GTO 24
304 ST- 34

305♦LBL 98
306 200
307 *
308 40
309 +
310 STO 35
311 RCL IND 35
312 1000000
313 /
314 INT
315 RCL 31
316 X>Y?
317 GTO 23
318 STO 32
319 RCL IND 35
320 1000000
321 /
322 FRC
323 RCL 32
324 +
325 1000000
326 *
327 STO IND 35
328 GTO 23

329♦LBL 24
330 DSE 33
331 GTO 22

332♦LBL "CPMPR"
333 40
334 STO 24
335 "START ACT.#"
336 98
337 ENTER↑
338 1
339 XEQ "*I"
340 2
341 *
342 40
343 +
344 STO 30
345 1
346 +
347 STO 31
348 FC? 55
349 GTO 30
350 "SEE KEY"
351 XEQ "*Y"
352 FS? 10
353 XEQ 49
354 PRBUF

355♦LBL 30
356 CF 15
357 FS? 55
358 PRBUF
359 RCL IND 30
360 X=0?
361 GTO 45
362 INT
363 1000
364 /
365 STO 33
366 FRC
367 ST- 33
368 STO 36
369 1000
370 ST/ 33
371 RCL 33
372 FRC
373 ST- 33
374 STO 37
375 1000
376 ST* 36
377 ST* 37
378 RCL 36
379 RCL 37
380 +
381 STO 38
382 RCL 33
383 RCL 36
384 -
385 STO 39
386 RCL IND 31
387 STO 34
388 100
389 ST/ 34
390 FC? 55
391 GTO 41
392 14
393 STO 41
394 XEQ 50
395 CLA
396 "TF="
397 ACA
398 CLA
399 RCL 39
400 RCL 37
401 -
402 CLA
403 ARCL X
404 ACA
405 CLA
406 X=0?
407 SF 15
408 FIX 3
409 PRBUF
410 3
411 STO 41
412 XEQ 50
413 "------"
414 ACA
415 ACA
416 ACA
417 PRBUF
418 2
419 STO 41
420 XEQ 50
421 "/"
422 ACA
423 1
424 STO 41
425 RCL 37
426 ACX
427 XEQ 54
428 -1
429 STO 41
```

Figure B5. (Cont'd).

```
430 RCL 40
431 XEQ 54
432 35
433 ACCHR
434 CLA
435 ARCL 40
436 ACA
437 2
438 STO 41
439 RCL 38
440 XEQ 54
441 RCL 38
442 ACX
443 CLA
444 "/"
445 ACA
446 PRBUF
447 1
448 STO 41
449 XEQ 50
450 ACA
451 6
452 STO 41
453 RCL 36
454 XEQ 54
455 RCL 36
456 ACX
457 6
458 SKPCHR
459 ACA
460 PRBUF

461♦LBL 33
462 2
463 SKPCHR
464 ACA
465 0
466 STO 41
467 RCL 39
468 ACX
469 XEQ 54
470 FS? 15
471 GTO 35
472 7
473 SKPCHR
474♦LBL 34
475 RCL 33
476 XEQ 54
477 RCL 33
478 ACX
479 ACA
480 PRBUF
481 GTO 36

482♦LBL 35
483 1
484 SKPCHR
485 CLA
486 "******"
487 ACA
488 CLA
489 "/"
490 GTO 34

491♦LBL 36
492 CLA
493 1
494 SKPCHR
495 "------"
496 ACA
497 ACA
498 ACA
499 PRBUF
500 1
501 ST+ 40
502 2
503 ST+ 30
504 ST+ 31
505 GTO 30

506♦LBL 41
507 RCL 40
508 FIX 0
509 "ACT#"
510 XEQ "*O"
511 RCL 36
512 "DURATION"
513 XEQ "*O"
514 RCL IND 31
515 STO 34
516 X≠0?
517 XEQ 44
518 RCL 37
519 "ES"
520 XEQ "*O"
521 RCL 38
522 "EF"
523 XEQ "*O"
524 RCL 39
525 "LS"
526 XEQ "*O"
527 RCL 33
528 "LF"
529 XEQ "*O"
530 RCL 39
531 RCL 37
532 -
533 "TF"
534 XEQ "*O"
535 1
536 ST+ 40
537 2
538 ST+ 30
539 ST+ 31
540 GTO 30

541♦LBL 44
542 100
543 ST/ 34
544 RCL 34
545 X=0?
546 RTN
547 FRC
548 ST- 34
549 100
550 *
551 "PRED.ACT"
552 XEQ "*O"
553 GTO 44

554♦LBL 45
555 XEQ "*P"
556 FIX 4
557 STOP
558 GTO "CPM"
```

Figure B5. (Cont'd).

```
559♦LBL 49
560 CLA
561 6
562 SKPCHR
563 "------"
564 ACA
565 ACA
566 ACA
567 PRBUF
568 5
569 SKPCHR
570 "/ES"
571 ACA
572 2
573 SKPCHR
574 "ACT#"
575 ACA
576 3
577 SKPCHR
578 "EF/"
579 ACA
580 PRBUF
581 4
582 SKPCHR
583 "/"
584 ACA
585 4
586 SKPCHR
587 "DURATI"
588 ACA
589 "ON"
590 ACA
591 4
592 SKPCHR
593 "/"
594 ACA
595 PRBUF
596 3
597 SKPCHR
598 "/LS"
599 ACA
600 11
601 SKPCHR
602 "LF/"
603 ACA
604 PRBUF
605 1
606 SKPCHR
607 "------"
608 ACA
609 ACA
610 ACA
611 PRBUF
612 CLA
613 RTN

614♦LBL 50
615 FIX 3
616 RCL 34
617 FRC
618 ST- 34
619 100
620 ST/ 34
621 *
622 FIX 0
623 X=0?
624 GTO 51
625 CLA
626 ARCL X
627 ACA
628 CLA
629 "/"
630 10
631 X<=Y?
632 GTO 52
633 1
634 GTO 53

635♦LBL 51
636 3
637 GTO 53

638♦LBL 52
639 0

640♦LBL 53
641 RCL 41
642 +
643 SKPCHR
644 RTN

654♦LBL 54
646 100
647 X<=Y?
648 GTO 55
649 RDN
650 10
651 X<=Y?
652 GTO 56
653 2
654 GTO 57

655♦LBL 55
656 0
657 GTO 57

658♦LBL 56
659 1

660♦LBL 57
661 RCL 41
662 +
663 SKPCHR
664 RTN
665 "15/3/82"
666 .END.
```

Figure B5. (Cont'd).

Table B1

"CPM" Program Example -- Without Printer

Step	Press	Resulting Display
1	XEQ	
	ALPHA	
	CPM	
	ALPHA	TOTAL # ACTIVITIES = ?
2	5 R/S	ACTIVITY # = ?
3	1 R/S	DURATION = ?
4	15 R/S	PRED. ACT = ?
5	0 R/S	ACTIVITY # = ?
6	2 R/S	DURATION = ?
7	20 R/S	PRED. ACT = ?
8	1 R/S	PRED. ACT = ?
9	0 R/S	ACTIVITY # = ?
10	3 R/S	DURATION = ?
11	18 R/S	PRED. ACT = ?
12	1 R/S	PRED. ACT = ?
13	0 R/S	ACTIVITY # = ?
14	4 R/S	DURATION = ?
15	18 R/S	PRED. ACT = ?
16	3 R/S	PRED. ACT = ?
17	0 R/S	ACTIVITY # = ?
18	5 R/S	DURATION = ?
19	20 R/S	PRED. ACT = ?
20	2 R/S	PRED. ACT = ?
21	4 R/S	PRED. ACT = ?
22	0 R/S	ACTIVITY # = ?
23	0 R/S	ANY CHANGES (Y/N)?
24	N R/S	ENTER ENDING ACTIVITIES
		END ACT. # = ?
25	5 R/S	END ACT. # = ?
26	0 R/S	START ACT. # = ?
27	1 R/S	ACT # = 1.
28	R/S	DURATION = 15.
29	R/S	ES = 0.
30	R/S	EF = 15.
31	R/S	LS = 0.
32	R/S	LF = 15.
33	R/S	TF = 0
34	R/S	ACT # = 2.
35	R/S	DURATION = 20.
36	R/S	PRED. ACT = 1.
37	R/S	ES = 15.
38	R/S	EF = 35.
39	R/S	LS = 31.
40	R/S	LF = 51.
41	R/S	TF = 16.
42	R/S	ACT # = 3.
43	R/S	DURATION = 18.

Table B1 (Cont'd)

Step	Press	Resulting Display
44	R/S	PRED. ACT = 1.
45	R/S	ES = 15.
46	R/S	EF = 33.
47	R/S	LS = 15.
48	R/S	LF = 33.
49	R/S	TF = 0.
50	R/S	ACT # = 4.
51	R/S	DURATION = 18.
52	R/S	PRED. ACT = 3.
53	R/S	ES = 33.
54	R/S	EF = 51.
55	R/S	LS = 33.
56	R/S	LF = 51.
57	R/S	TF = 0.
58	R/S	ACT # = 5.
59	R/S	DURATION = 20.
60	R/S	PRED. ACT = 2.
61	R/S	PRED. ACT = 4.
62	R/S	ES = 51.
63	R/S	EF = 71.
64	R/S	LS = 51.
65	R/S	LF = 71.
66	R/S	TF = 0.
67	R/S	END PROGRAM

Table B2

"CPM" Program -- Special Register Uses

Register Number	Register Contents
	LBL 00 *
30	Loop control value and indirect storage for preceding activities
31	Activity number
32	Duration of activity
	LBL 01
33-37	Preceding activities
	LBL 04
30	Coded activity register value for activity durations (or) coded activity register value for preceding activities
31	Loop control value for preceding activities
32	1
(ACT # x 2)+40	Duration of activity
	LBL 05
32	Preceding activities storage factor
(ACT # x 2)+41	Preceding activities for one activity
	LBL 09
30	1
32	Loop control value for ending activities
36	Loop control value for ending activities

*See Table B3 for description of labels.

Table B2 (Cont'd)

Register Number	Register Contents
	LBL 10
30	Ending activity storage factor
36	Loop control value for ending activity values
37-40	Ending activity values
	LBL 11
31	Coded activity register value for activity number 1
41	Loop control value
	LBL 12
30	0
32	Preceding activities
35	0
	LBL 14
32	Preceding activities
33	Duration of preceding activities
34	Total duration for preceding activities
	LBL 15
35	Largest duration for preceding activities
	LBL 16
31	Coded activity register value containing activity durations (or) coded activity register value containing preceding activities
41	Loop control value (or) total number of activities
(ACT # x 2)+40	Contains value XXX,YYY where XXX = largest duration of preceding activities and YYY = duration of activity

Table B2 (Cont'd)

Register Number	Register Contents
	LBL 17
33	Loop control value and storage for ending activities
	LBL 18
30	Ending activities
34	Loop control value for ending activities
	LBL 20
30	Ending activity storage factor
	LBL 21
33	Loop control value and storage
	LBL 22
30	Late finish, early start, and duration for activity
31	Duration of activity (or) late start value
34	Preceding activities
	LBL 23
34	Preceding activities
	LBL 30
33	Late finish time for activity
34	Preceding activities
36	Duration of activity
37	Early start time for activity
38	Early finish time for activity
39	Late start time for activity
41	Number of spaces to be skipped

Table B2 (Cont'd)

Register Number	Register Contents
	LBL 33
41	Number of spaces to be skipped
	LBL 36
30	Coded activity register containing LF, ES, and duration
31	Coded activity register containing preceding activities
40	Activity number
	LBL 41
30	Coded activity register containing LF, ES, and duration
31	Coded activity register containing preceding activities
40	Activity number
	LBL 44
34	Preceding activities
	LBL 50
34	Preceding activities
	LBL 98
32	Late finish value
35	Coded activity register value
(ACT # x 2)+40	Stores the value XXX,YYY,ZZZ where XXX = LF, YYY = ES, and ZZZ = duration for each activity
	LBL 99
42-[(TOTAL # ACT x 2)+40]	Adds 999,000,000 to XXX,YYY value

Table B2 (Cont'd)

Register Number	Register Contents
	LBL a
30	Ending activities
31	Greatest duration of preceding activities
32	Greatest late finish time for ending activities
35	Duration of activity
	LBL b
30	Ending activities multiplication factor
(ACT # x 2)+40	Contains XXX,YYY,ZZZ where XXX = LF, YYY = ES, and ZZZ = duration
	LBL CPM
41	Total number of activities
	LBL CPMPR
30	Coded activity register
31	Coded activity register
40	Starting activity number

Table B3

"CPM" Program -- Functions, By Label

Label	Purpose
00	Inputs activity number and duration for each activity
01	Inputs preceding activities
04	Stores durations in coded activity registers
05	Stores preceding activities in coded activity registers
08	Determines if any changes are required
09	Intermediate calculations; displays "ENTER ENDING ACTIVITIES"
10	Inputs/stores ending activity numbers
11	Computes/stores loop control value
12	Stores preceding activities in R_{32}
14	Determines the greatest duration of combined preceding activities
15	Stores largest duration of combined preceding activities
16	Combines/stores the duration of an activity and early start
17	Stores loop control value
18	Stores ending activities; stores loop control value
19	Intermediate calculation
20	Intermediate calculation
21	Computes/stores loop control value
22	Computes/stores (late finish - duration) for each activity
23	Intermediate calculation
24	Intermediate calculation
30	Computes/stores output values
33	Intermediate calculation
34	Outputs late finish
35	Displays "******" for activities on the critical path

Table B3 (Cont'd)

Label	Purpose
36	Outputs bottom of CPM box; computes/stores new activity number and coded activity registers
41	Outputs activity number, duration, ES, EF, LS, LF and TF when printer is not attached
44	Outputs preceding activities when printer is not attached
45	Displays "END PROGRAM"
49	Outputs key for CPM display
50	Outputs preceding activities on printer
51	Intermediate storage value
52	Intermediate storage value
53	Computes number of spaces to be skipped
54	Intermediate calculation
55	Intermediate storage value
56	Intermediate storage value
57	Computes number of spaces to be skipped
98	Computes/stores XXX,YYY,ZZZ for each activity, where XXX = LF, YYY = ES, and ZZZ = duration
99	Adds 999,000,000 to all coded activity registers
a	Determines/stores late finish value for ending activities
b	Adds XXX,000,000 to ending activities where XXX = LF
CPM	Marks beginning program; checks size allocation; inputs number of activities
CPMPR	Inputs starting activity number; determines if key is to be output

APPENDIX C:
"CRATER" PROGRAM DETAILS

This appendix provides detailed information about the "CRATER" program. Figure C1 shows the typical sequence of events and the options that a user encounters when executing this program. Table C1 is an example of the specific steps that must be followed to solve a problem when one uses the HP-41 calculator without a printer attached. In this example, the program determines the amount of TNT, and the number of boreholes and 40-lb cratering charges needed to blast a 41-ft-long deliberate crater.

"CRATER" presents the user a menu for types of craters. The menu order is "hasty," "deliberate," and "relieved face." In step 4 of Table C1, the user has to press the R/S key to restart the program after only part of the message has been displayed. If a printer were attached and set to the "NORM" printer mode, the program would advance automatically. Steps 7 through 13 of the same example would also be executed automatically if a printer were attached. Figure C2 shows three examples of using "CRATER" program with a printer attached.

Abbreviations used in "CRATER" are:

Symbol	Meaning
CHG	Charge
EXPLO	Explosive(s)
FT	Feet
LBS	Pounds
(Y/N)	(Yes/No)
#	Number
Σ	Sum of (total)

There are two sets of operating limits for input variables in the program:

Variable	Units	Minimum	Maximum
Crater length	Feet	12	999
Crater depth	Feet	7.5	15

A minimum length of 12 ft is required to insure that the formula for the number of holes, $N = (\frac{L-16}{5}) + 1$, is always positive. The maximum crater depth was determined by assuming that the maximum practical borehole depth was 10 ft, and multiplying this by 1.5 to get 15 ft for the maximum crater depth. Algorithms used in the program were taken from FM 5-34 (pp 28-32) and from FM 5-25.[2] Critical assumptions and formulas are as follows for each crater option:

---

[2] Explosives and Demolitions, FM 5-25 (HQ, DA, February 1971), pp 3-21 through 3-25.

1. For hasty craters:

a. Holes of equal depth are spaced at 5-ft intervals and are packed with 10 lb of explosives per foot of depth.

b. Crater depth is 1.5 times depth of borehole

c. Minimum borehole depth is 5 ft

d. Maximum practical borehole depth is 10 ft

e. Number of holes = N = $(\frac{L-16}{5}) + 1$, rounded up; L = crater length, in feet.

2. For deliberate craters:

a. Holes are 5 ft apart, with both end holes 7-ft deep; depth of holes alternate, but no 5-ft holes are adjacent

b. 80 lb of explosives in 7-ft holes; 40 lb in 5-ft holes

c. Number of holes: same as for hasty crater.

3. For relieved face craters:

a. Number of 5-ft holes (friendly side): $N_5 = (\frac{L-10}{7}) + 1$, rounded up

b. Number of 4-ft holes (enemy side): $N_4 = N_5 - 1$

c. 40 lb of explosive in 5-ft holes; 30 lb in 4-ft holes.

4. General: if a hole has only one cratering charge in it, an extra pound of TNT must be provided to meet the double-priming requirement.

"CRATER" uses registers 31 through 39 to store the values described in Table C2. If cratering charges are used, the program sets general purpose flag "00".

Table C3 describes the general function of each part of the program, by label. Figure C3 is a label wiring diagram showing how the different parts of the program relate to each other. A circular loop on the diagram indicates a return to the same label. This happens, for example, when the user does not select any of the choices in a menu and the menu is presented again. One other convention used in the wiring diagram is a two-headed arrow pointing both to and from a local subroutine used by the "CRATER" program only. This indicates the program executes that subroutine, then returns to the main program. Global subroutines, used by all the major application programs on the MILENG1/UTIL module, are not shown on the wiring diagram, but are described separately in Appendix G.

Figure C4 presents a detailed flowchart of the "CRATER" program, and Figure C5 lists the program steps.

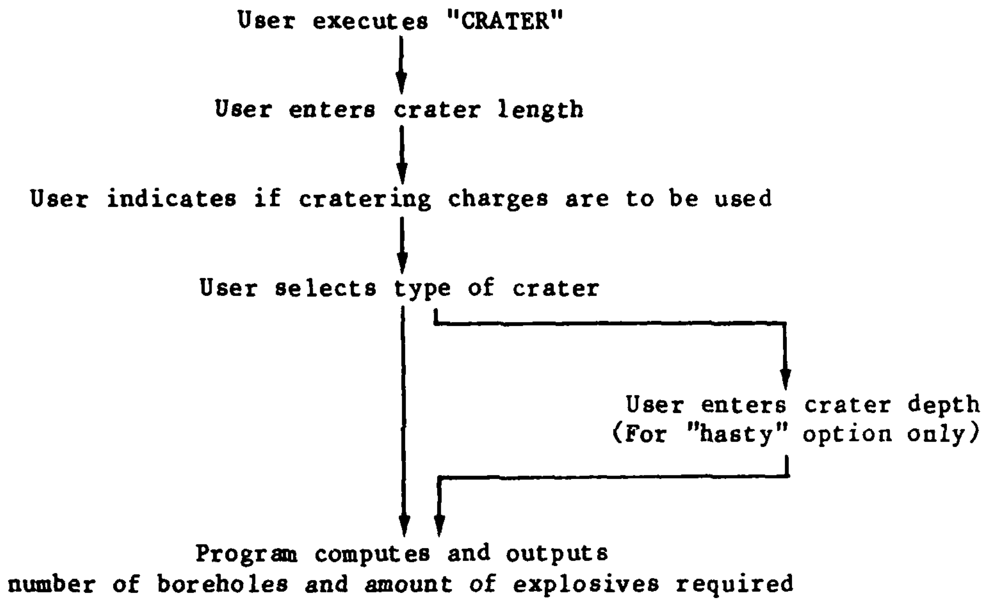


Figure C1. "CRATER" program sequence of events.

```
              XEQ "CRATER"
CRATER LENGTH,(FT)=?
         41.0         RUN
USE CRATER CHARGE(Y/N)?
N                     RUN
CRATER TYPE:
HASTY(Y/N)?
Y                     RUN
CRATER DEPTH,(FT)=?
          7.5         RUN

#HOLES=6.
HOLE DEPTH,FT=5.0
EXPLOSIVE, LBS/HOLE=50.

EXPLO,ΣLB=300.
ALSO:  NEED SHAPE CHARGES
TO BLAST BOREHOLES!
END PROGRAM
```

```
              XEQ "CRATER"
CRATER LENGTH,(FT)=?
         41.0         RUN
USE CRATER CHARGE(Y/N)?
Y                     RUN
CRATER TYPE:
HASTY(Y/N)?
N                     RUN
DELIBERATE(Y/N)/?
Y                     RUN

#7FT.HOLES=4.
#5FT.HOLES=2.
CRATER CHG=10.
PRIMER:TNT,LBS=2.

EXPLO, ΣLB=402.
ALSO:  NEED SHAPE CHARGES
TO BLAST BOREHOLES!
END PROGRAM
```

```
              XEQ "CRATER"
CRATER LENGTH,(FT)=?
         41.0         RUN
USE CRATER CHARGE(Y/N)?
Y                     RUN
CRATER TYPE:
HASTY(Y/N)?
N                     RUN
DELIBERATE(Y/N)?
N                     RUN
RELIEVED FACE(Y/N)?
Y                     RUN

FRIEND SIDE:
#5FT.HOLES=6.
#CRATER CHG=6.
PRIMER:TNT,LBS=6.

ENEMY SIDE:
#4FT.HOLES=5.
TNT,LBS=150.

EXPLO,ΣLB=396.
ALSO:  NEED SHAPE CHARGES
TO BLAST BOREHOLES!
END PROGRAM
```

Figure C2. "CRATER" program examples -- with printer.

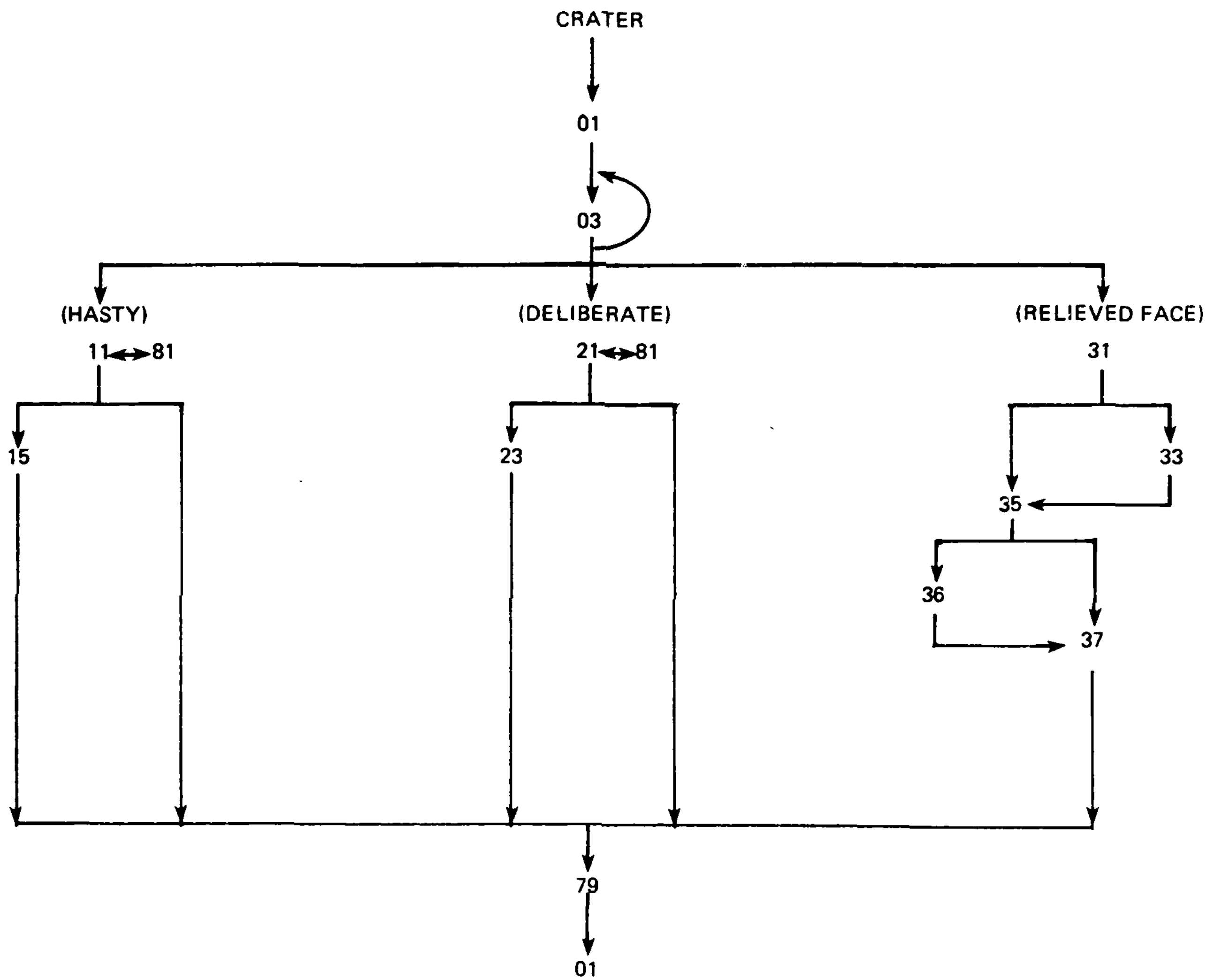


Figure C3. "CRATER" program label wiring diagram.

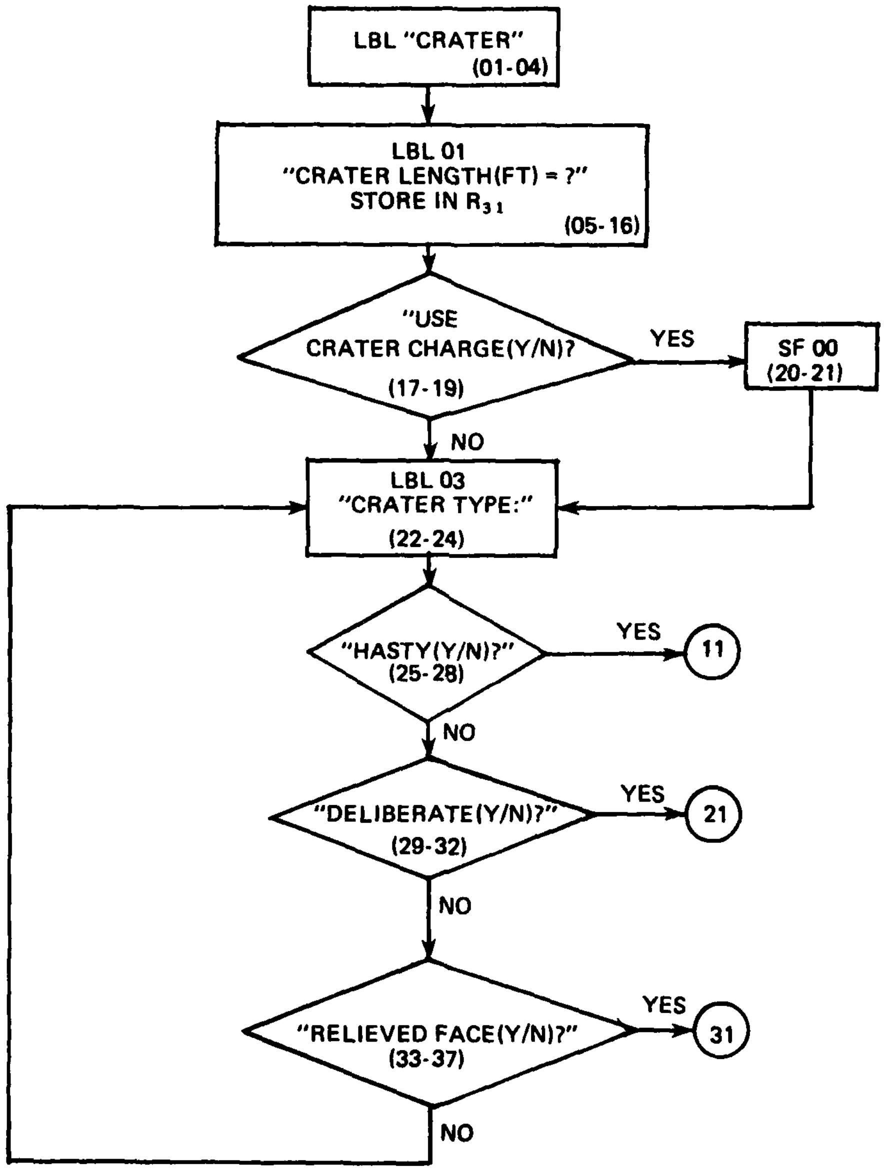


Figure C4. "CRATER" program -- detailed flowchart.

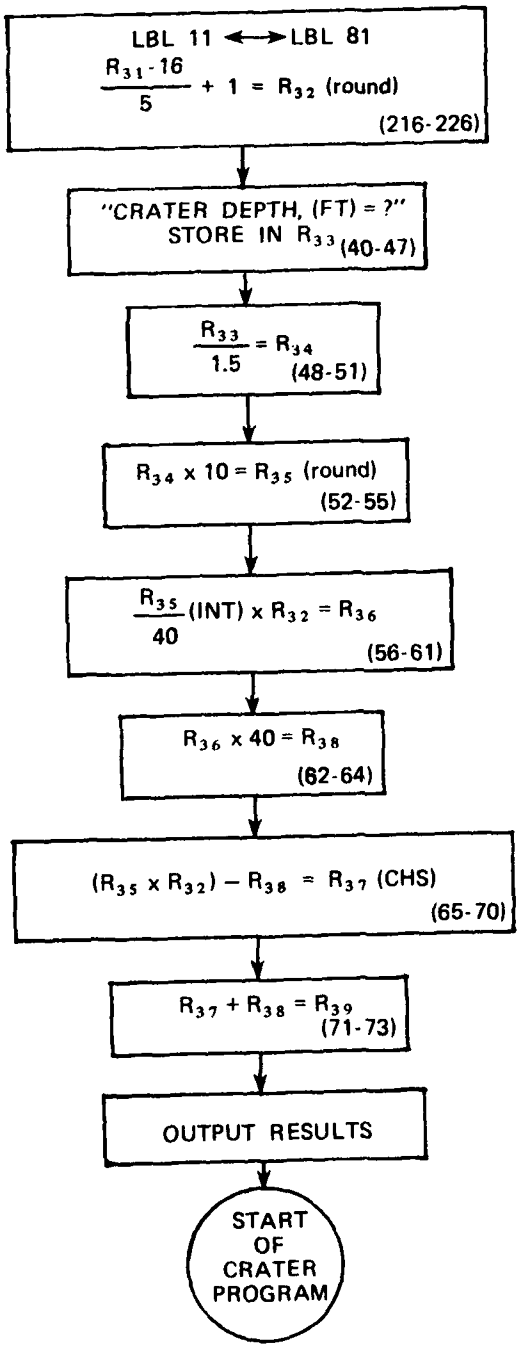


Figure C4. (Cont'd).

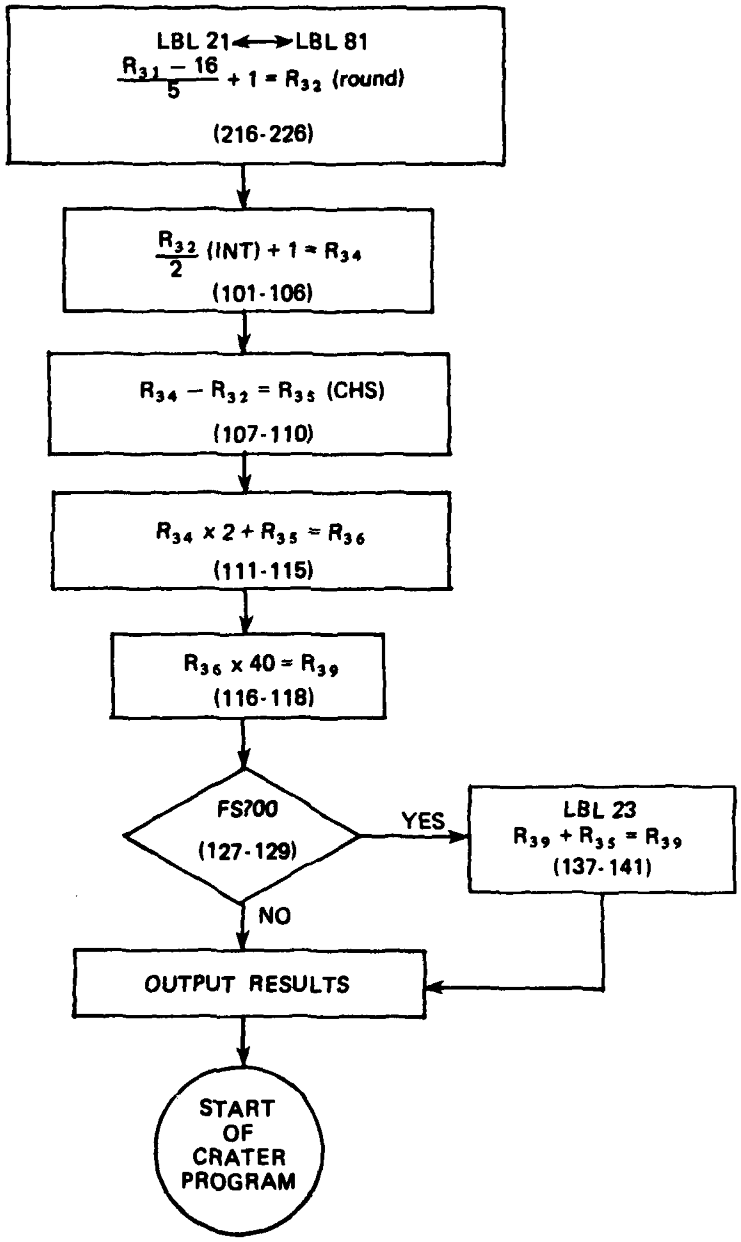


Figure C4. (Cont'd).

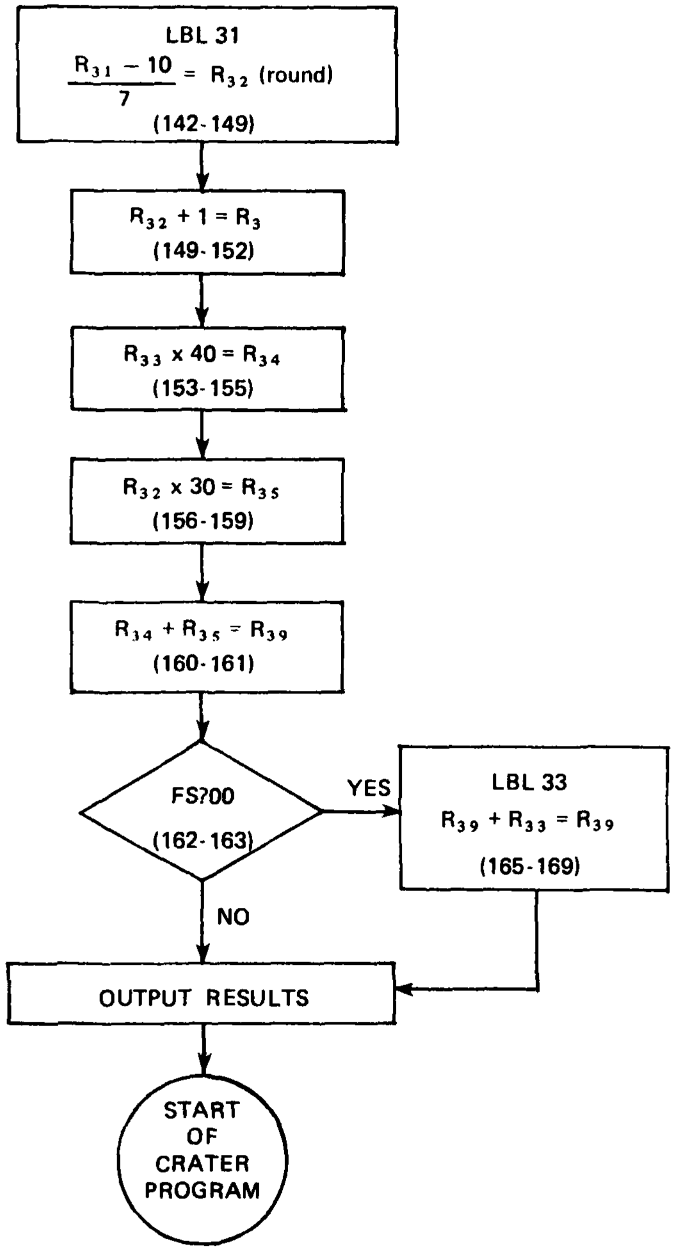


Figure C4. (Cont'd).

```
01♦LBL "CRATER"
02 40
03 XEQ "*S"
04 CF 00

05♦LBL 01
06 FIX 1
07 31.039
08 XEQ "*C"
09 31
10 STO 24
11 "CRATER LENGTH,("
12 "⊢FT)"
13 999
14 ENTER↑
15 3
16 XEQ "*I"
17 "USE CRATER CHAR"
18 "⊢GE"
19 XEQ "*Y"
20 FS? 10
21 SF 00

22♦LBL 03
23 "CRATER TYPE:"
24 XEQ "*D"
25 "HASTY"
26 XEQ "*Y"
27 FS? 10
28 GTO 11
29 "DELIBERATE"
30 XEQ "*Y"
31 FS? 10
32 GTO 21
33 "RELIEVED FACE"
34 XEQ "*Y"
35 FS? 10
36 GTO 31
37 GTO 03

38♦LBL 11
39 XEQ 81
40 33
41 STO 24
42 "CRATER DEPTH,(F"
43 "⊢T)"
44 15
45 ENTER↑
46 7.5
47 XEQ "*I"
48 RCL 33
49 1.5
50 /
51 STO 34
52 10
53 *
54 XEQ "*R"
55 STO 35
56 40
57 /
58 INT
59 RCL 32
60 *
61 STO 36
62 40
63 *
64 STO 38
65 RCL 35
66 RCL 32
67 *
68 -
69 CHS
70 STO 37
71 RCL 38
72 +
73 STO 39
74 ADV
75 FIX 0
76 "#HOLES"
77 RCL 32
78 XEQ "*O"
79 FIX 1
80 "HOLE DEPTH,FT"
81 RCL 34
82 XEQ "*O"
83 FIX 0
84 "EXPLOSIVE, LBS/"
85 "⊢HOLE"
86 RCL 35
87 XEQ "*O"
88 FS? 00
89 GTO 15
90 GTO 79

91♦LBL 15
92 "#CRATER CHG"
93 RCL 36
94 XEQ "*O"
95 "TNT,LBS"
96 RCL 37
97 XEQ "*O"
98 GTO 79

99♦LBL 21
100 XEQ 81
101 2
102 /
103 INT
104 1
105 +
106 STO 34
107 RCL 32
108 -
109 CHS
110 STO 35
111 RCL 34
112 2
113 *
114 +
115 STO 36
116 40
117 *
118 STO 39
119 ADV
120 FIX 0
121 "#7FT.HOLES"
122 RCL 34
123 XEQ "*O"
124 "#5FT.HOLES"
125 RCL 35
126 XEQ "*O"
127 FS? 00
128 GTO 23
129 GTO 79

130♦LBL 23
131 "#CRATER CHG"
132 RCL 36
133 XEQ "*O"
134 "PRIMER:TNT,LBS"
135 RCL 35
136 XEQ "*O"
137 RCL 39
138 RCL 35
139 +
140 STO 39
141 GTO 79

142♦LBL 31
143 RCL 31
144 10
145 -
146 7
147 /
```

Figure C5. "CRATER" program listing.

```
148 XEQ "*R"
149 STO 32
150 1
151 +
152 STO 33
153 40
154 *
155 STO 34
156 RCL 32
157 30
158 *
159 STO 35
160 +
161 STO 39
162 FS? 00
163 GTO 33
164 GTO 35

165♦LBL 33
166 RCL 39
167 RCL 33
168 +
169 STO 39

170♦LBL 35
171 ADV
172 FIX 0
173 "FRIEND SIDE:"
174 XEQ "*D"
175 "#5FT.HOLES"
176 RCL 33
177 XEQ "*O"
178 FS? 00
179 GTO 36
180 "TNT,LBS"
181 RCL 34
182 XEQ "*O"
183 GTO 37

184♦LBL 36
185 "#CRATER CHG"
186 RCL 33
187 XEQ "*O"
188 "PRIMER:TNT,LBS"
189 RCL 33
190 XEQ "*O"

191♦LBL 37
192 ADV
193 "ENEMY SIDE:"
194 XEQ "*D"
195 "#4FT.HOLES"
196 RCL 32
197 XEQ "*O"
198 "TNT,LBS"
199 RCL 35
200 XEQ "*O"

201♦LBL 79
202 ADV
203 "EXPLO,ΣLB"
204 RCL 39
205 XEQ "*O"
206 "ALSO:  NEED SHAP"
207 "⊢E CHARGES"
208 XEQ "*D"
209 "TO BLAST BOREHO"
210 "⊢LES!"
211 XEQ "*D"
212 CF 00
213 XEQ "*P"
214 STOP
215 GTO 01

216♦LBL 81
217 RCL 31
218 16
219 -
220 5
221 /
222 1
223 +
224 XEQ "*R"
225 STO 32
226 RTN
227 "2/2/82"
228 .END.
```

Figure C5. (Cont'd).

Table C1

"CRATER" Program Example -- Without Printer

Step	Press	Resulting Display
1	XEQ	
	ALPHA	
	CRATER	
	ALPHA	CRATER LENGTH, (FT) = ?
2	41 R/S	USE CRATER CHARGE (Y/N)?
3	Y R/S	CRATER TYPE:
4	R/S	HASTY (Y/N)?
5	N R/S	DELIBERATE (Y/N)?
6	Y R/S	# 7 FT. HOLES = 4.
7	R/S	# 5 FT. HOLES = 2.
8	R/S	# CRATER CHG = 10.
9	R/S	PRIMER: TNT, LBS = 2.
10	R/S	EXPLO, ΣLB = 402.
11	R/S	ALSO: NEED SHAPE CHARGES
12	R/S	TO BLAST BOREHOLES!
13	R/S	END PROGRAM

Table C2

"CRATER" Program -- Special Register Uses

Register Number	Option Code*	Register Contents
31	H,D,R -	Crater length in feet
32	H,D -	Number of holes
	R -	Number of holes in enemy row (4 foot holes)
33	H -	Crater depth in feet
	R -	Number of holes in friendly row (5-ft holes); number of cratering charges; pounds of TNT primer
34	H -	Hole depth in feet
	D -	Number of 7-ft holes
	R -	Pounds of explosive for friendly row (5-ft holes)
35	H -	Pounds of explosive per hole
	D -	Number of 5-ft holes; pounds of TNT primer
	R -	Pounds of explosive for enemy row (4-ft holes)
36	H,D -	Number of cratering charges required
37	H -	Pounds of TNT required if cratering charges are used
38	H -	Intermediate value
39	H,D,R -	Total pounds of explosive required

*H = hasty; D = deliberate; R = relieved face

Table C3

"CRATER" Program -- Functions, By Label

Label	Purpose
CRATER	Marks beginning of program; checks size allocation
01	Clears registers; inputs crater length in feet; determines if cratering charges are to be used
03	Presents crater-type menu
11	Inputs crater depth; computes/outputs values for a hasty crater
15	Outputs certain values for a hasty crater (if cratering charges are used)
21	Computes/outputs values for a deliberate crater
23	Computes/outputs certain values for a deliberate crater (if cratering charges are used)
31	Computes values for a relieved face crater
33	Computes certain values for a relieved face crater (if cratering charges are used)
35	Outputs friendly side values for a relieved face crater
36	Outputs friendly side values for a relieved face crater (if cratering charges are used)
37	Outputs enemy side values for a relieved face crater
79	Outputs total pounds of explosive required; displays shape charge message; advises of program end
Local Sub-Routine	
81	Computes number of holes for hasty and deliberate craters

APPENDIX D:
"DEMO" PROGRAM DETAILS

This appendix provides detailed information about the "DEMO" program. Figure D1 shows the typical sequence of events and the options that a user encounters when executing this program. Table D1 is an example of the specific steps that must be followed to solve a problem when one uses the HP-41 calculator without a printer attached. In this example, the program determines the amount of explosive, the number of explosive units and charges, and the minimum safe distance required to breach a reinforced concrete wall that is 7-1/2-ft thick and 42-ft long using ground-placed, tamped charges.

"DEMO" presents the user four menus: explosive types, applications, charge placements, and steel types. The order of the first menu is TNT, M112 C4(1.25 lb), M5A1 C4(2.5 lb), Dynamite M1, Tetrytol, M118 Sheet (0.5 lb), and M186 Roll (25 lb). The next menu is "cut timber," "cut steel." and "breach wall." If the cut timber application option is selected, the next menu is charge placement. in this order: "abatis," "external." then "internal." The last menu is only presented in "cut steel" applications; the menu order is "RR rails." "round steel shapes," "structural steel." then "carbon steel rods." In step 2 of Table D1. the user has to press the R/S key to restart the program after only part of the message has been displayed. If a printer were attached and set to the "NORM" printer mode, the program would advance automatically, and steps 17 through 20 would be executed automatically. Figure D2 shows three examples of using "DEMO" program with a printer attached. Abbreviations used in "DEMO" are:

Symbol	Meaning	Symbol	Meaning
BREACH	Breaching	M	Meter
C	Tamping factor	MIN	Minimum
CONCR	Concrete	REINF	Reinforced
DEMO	Demolition	REQD	Required
DIA	Diameter	RR	Railroad
DIST	Distance	SECT	Section
EXPLO	Explosive	STL	Steel
EXTERN	External	STR.STL.	Structural steel
FT	Feet	SQ. IN	Square inches
HT	Height	TAB	Table
IN	Inches	X-SECT	Cross-sectional
INTERN	Internal	(Y/N)	(Yes/No)
K	Material factor	#	Number
LB	Pound(s)		

There are nine sets of operating limits for input variables in the program:

Variable	Units	Minimum	Maximum
Timber diameter	Inches	5	180
Rail height	Inches	1	9
Bar diameter	Inches	0	24
X-sect area	Sq in.	0	99
Section diameter	Inch	0	99
Material factor, K		0.07	1.76
Tamp factor, C		1	3.6
Barrier width	Feet	0	999
Breaching radius	Feet	0.1	99

Algorithms used in the program were taken from FM 5-34 (pp 24-29, and from FM 5-25 (pp 1-6 and 3-20). Critical assumptions and formulas are as follows for each application option:

1. For timber cutting applications:

a. Pounds of TNT (P) required for one charge, where D = diameter of timber in inches --

For external charge placement: $P = D^2/40$
For internal charge placement: $P = D^2/250$
For abatis charge placement: $P = D^2/50$.

b. Pounds of explosive = $\frac{P}{RE}$, where RE is the relative effectiveness factor

c. Number of explosive units = $\frac{\text{lb of explosive}}{\text{size of unit to be used}}$.

2. For steel cutting applications:

a. Pounds of TNT (P) re ·red for one charge, where D = diameter of timber in inches --

(1) For railroad rails (no effectiveness factor adjustment):
Rail height > 5 in., $P = 1$ lb
Rail height < 5 in., $P = 0.5$ lb
Railroad frogs, $P = 2$ lb

(2) For round steel shapes (no effectiveness factor adjustment):
Diameter < 1 in. $P = 1$ lb
1 in. < Diameter $\leq$ 2 in., $P = 2$ lb
Diameter > 2 in., $P = \frac{3}{8} \times (A)$*

(3) For structural steel sections: $P = \frac{3}{8} \times (A)$*

*A is cross-sectional area (sq in.).

(4) For high carbon steel rods, where D = diameter in inches: $P = D^2$.

b. Adjusted pounds of explosive = $\frac{P}{RE}$

c. Number of explosive units = $\frac{\text{lb of explosive}}{\text{size of unit to be used}}$.

3. For breaching wall applications:

a. Pounds of TNT (P) required for one charge, where R = breaching radius in feet, K = material factor, and C = tamping factor: $P = R^3 KC$

b. Adjusted pounds of explosive = $\frac{P}{RE}$

c. Number of charges (N), where W = barrier width and R = breaching radius in feet: $N = \frac{W}{2R}$

— if $N < 1\text{-}1/4$, then N = 1
— if $1\text{-}1/4, \leq N < 2\text{-}1/2$, then N = 2
— if $N \geq 2\text{-}1/2$, then round off.

d. Pounds of explosive required = (lb of explosive) x (no. of charges).

4. General: minimum safe distance (m) for personnel in the open

$$m=100 \sqrt[3]{\text{lb of explosive}}$$

"DEMO" uses registers 30 through 40 to store the values described in Table D2. The program sets general purpose flag "00" if the user elects to use the breach wall application.

Table D3 describes the general function of each part of the program, by label. Figure D3, a label wiring diagram, shows how the different parts of the program relate to each other. A circular loop on the diagram indicates a return to the same label. A two-headed arrow pointing to and from a label indicates the program executes that local subroutine, then returns to the main program. Global subroutines, used by all the major application programs on the MILENG1/UTIL module, are not shown on the wiring diagram, but are described separately in Appendix G.

Figure D4 presents a detailed flowchart of the "DEMO" program, and Figure D5 lists the program steps.

AD-A120 317 CONSTRUCTION ENGINEERING RESEARCH LAB (ARMY) CHAMPAIGN IL F/G 9/2
SOFTWARE DOCUMENTATION FOR MILENG1/UTIL READ-ONLY-MEMORY MODULE--ETC(U)
SEP 82 L A THOMAS, J M DEPONAI
UNCLASSIFIED CERL-TR-P-134 NL

AD A
120317

END
DATE
FILMED
11-82
DTIC

User executes "DEMO"

↓

User selects type of explosive

↓

User chooses application

↓

User enters appropriate data for the chosen application

↓

Program computes and outputs:
pounds of explosive required
number of explosive units
number of charges (for breaching application)
"in open" safe distance.

Figure D1. "DEMO" program sequence of events.

```
               XEQ "DEMO"
EXPLOSIVE TYPE:
TNT(Y/N)?
N                        RUN
M112 C4(1.25LB)(Y/N)?
Y                        RUN
APPLICATION:
CUT TIMBER(Y/N)?
Y                        RUN
TIMBER DIA.(IN)=?
            24.0         RUN
CHARGE PLACEMENT:
ABATIS(Y/N)?
N                        RUN
EXTERN.(Y/N)?
Y                        RUN

REQD.EXPLO,LBS=11.3
#EXPLO.UNITS=9.
IN OPEN,SAFE DIST,M=300.
END PROGRAM
```

```
               XEQ "DEMO"
EXPLOSIVE TYPE:
TNT(Y/N)?
Y                        RUN
BLOCKS,TNT,1LB(Y/N)?
N                        RUN
APPLICATION:
CUT TIMBER(Y/N)?
N                        RUN
CUT STEEL(Y/N)?
Y                        RUN
TYPE STEEL:
RR.RAIL(Y/N)?
Y                        RUN
RAIL.HT.(IN)=?
             5.0         RUN
RR.FROG(Y/N)?
N                        RUN

REQD.EXPLO,LBS=1.0
#EXPLO,UNITS=2.
IN OPEN,SAFE DIST,M=300.
END PROGRAM
```

```
               XEQ "DEMO"
EXPLOSIVE TYPE:
TNT(Y/N)?
N                        RUN
M112 C4(1.25LB)(Y/N)?
N                        RUN
M5A1 C4(2.5LB)(Y/N)?
N                        RUN
DYNAMITE,M1(Y/N)?
Y                        RUN
APPLICATION:
CUT TIMBER(Y/N)?
N                        RUN
CUT STEEL(Y/N)?
N                        RUN
BREACH(Y/N)?
Y                        RUN
TAB.2-3,FM5-34:
MATERIAL FACTOR,K=?
             .54         RUN
TAB.2-4,FM5-34:
TAMP FACTOR,C=?
             2.0         RUN
BARRIER WIDTH,(FT)=?
            42.0         RUN
BREACH, RADIUS,(FT)=?
             7.5         RUN

REQD.EXPLO,LBS=1,486.5
#EXPLO.UNITS=2,973.
#CHARGES=3.
 OPEN,SAFE DIST,M=1,141.
END PROGRAM
```

Figure D2. "DEMO" program examples -- with printer.

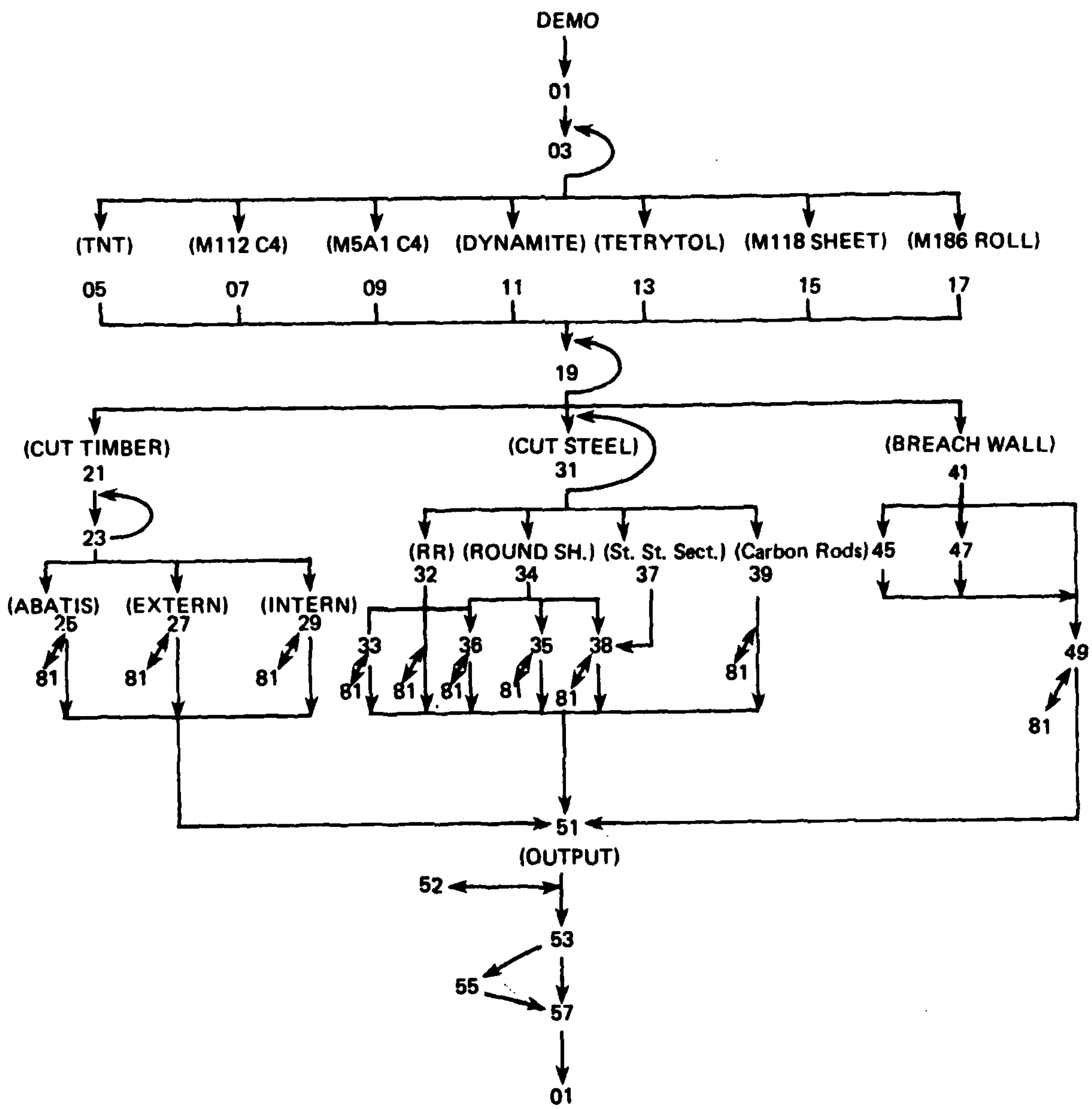


Figure D3. "DEMO" program label wiring diagram.

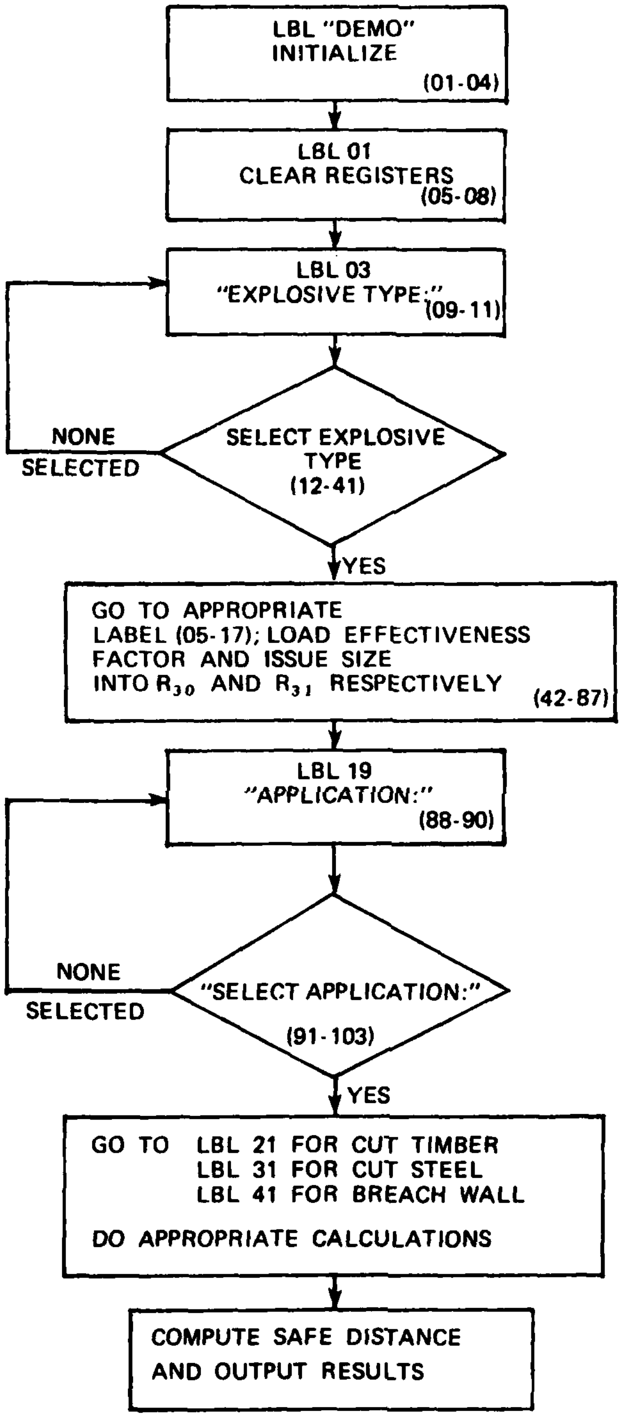


Figure D4. "DEMO" program -- detailed flowchart.

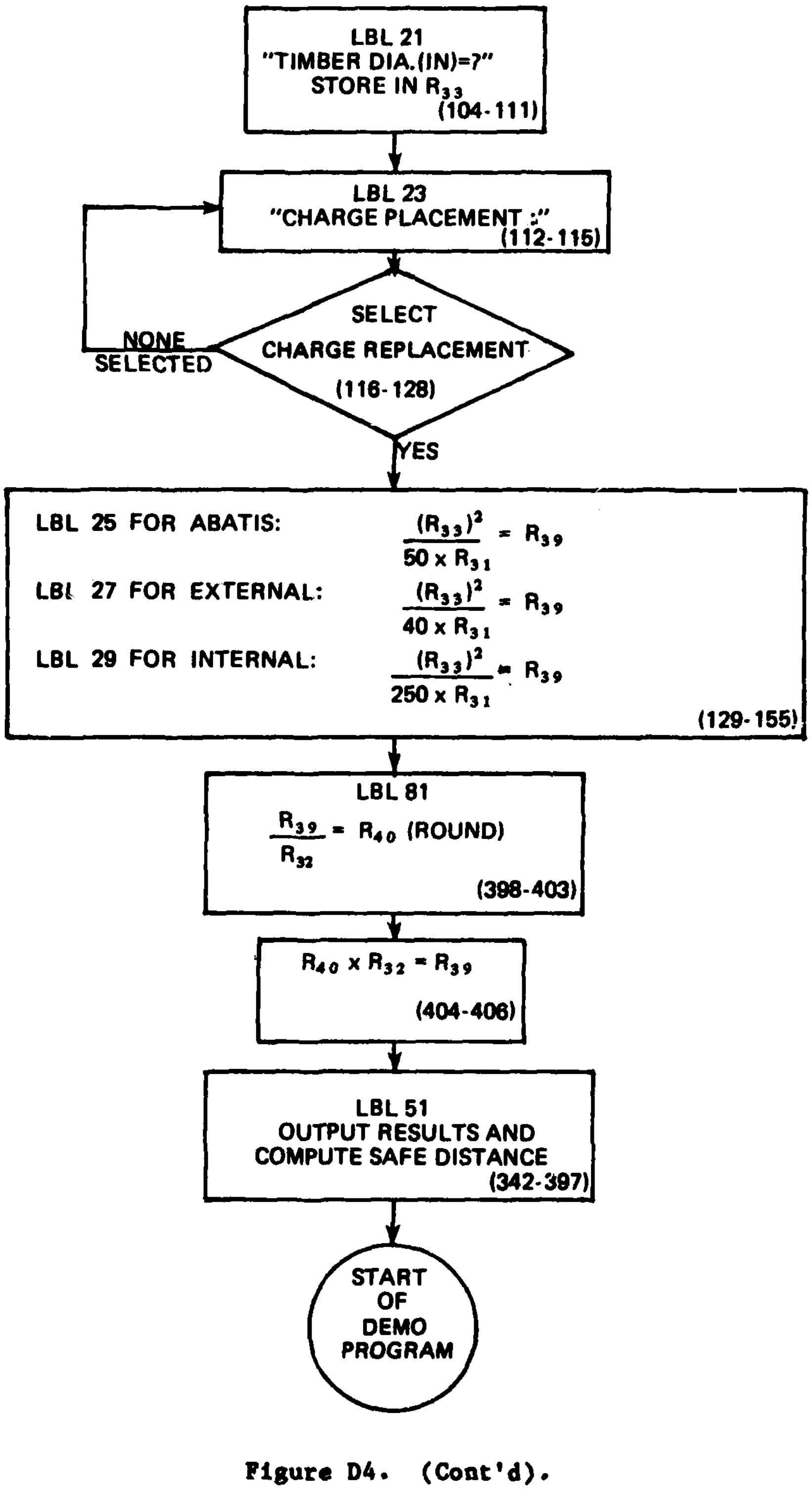


Figure D4. (Cont'd).

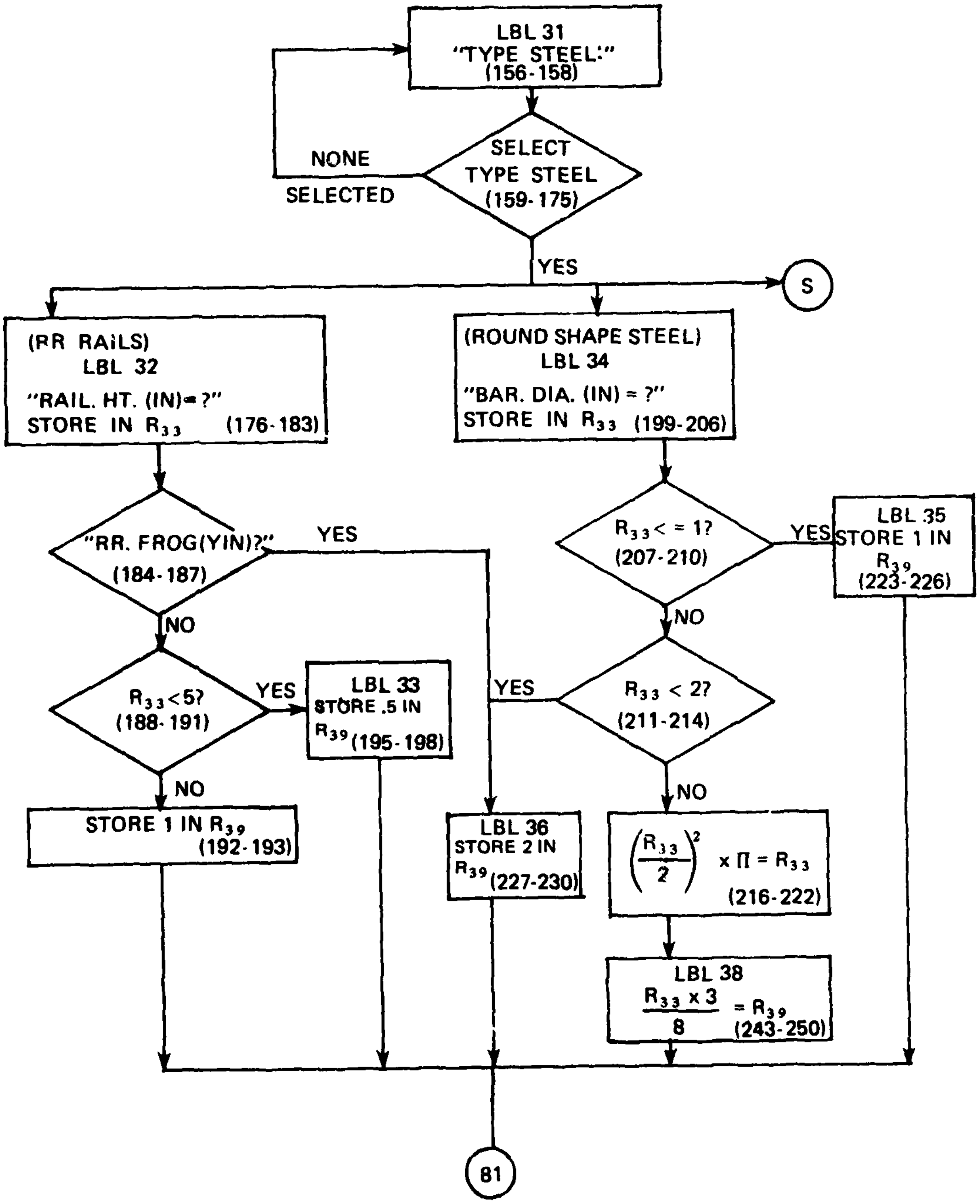


Figure D4. (Cont'd).

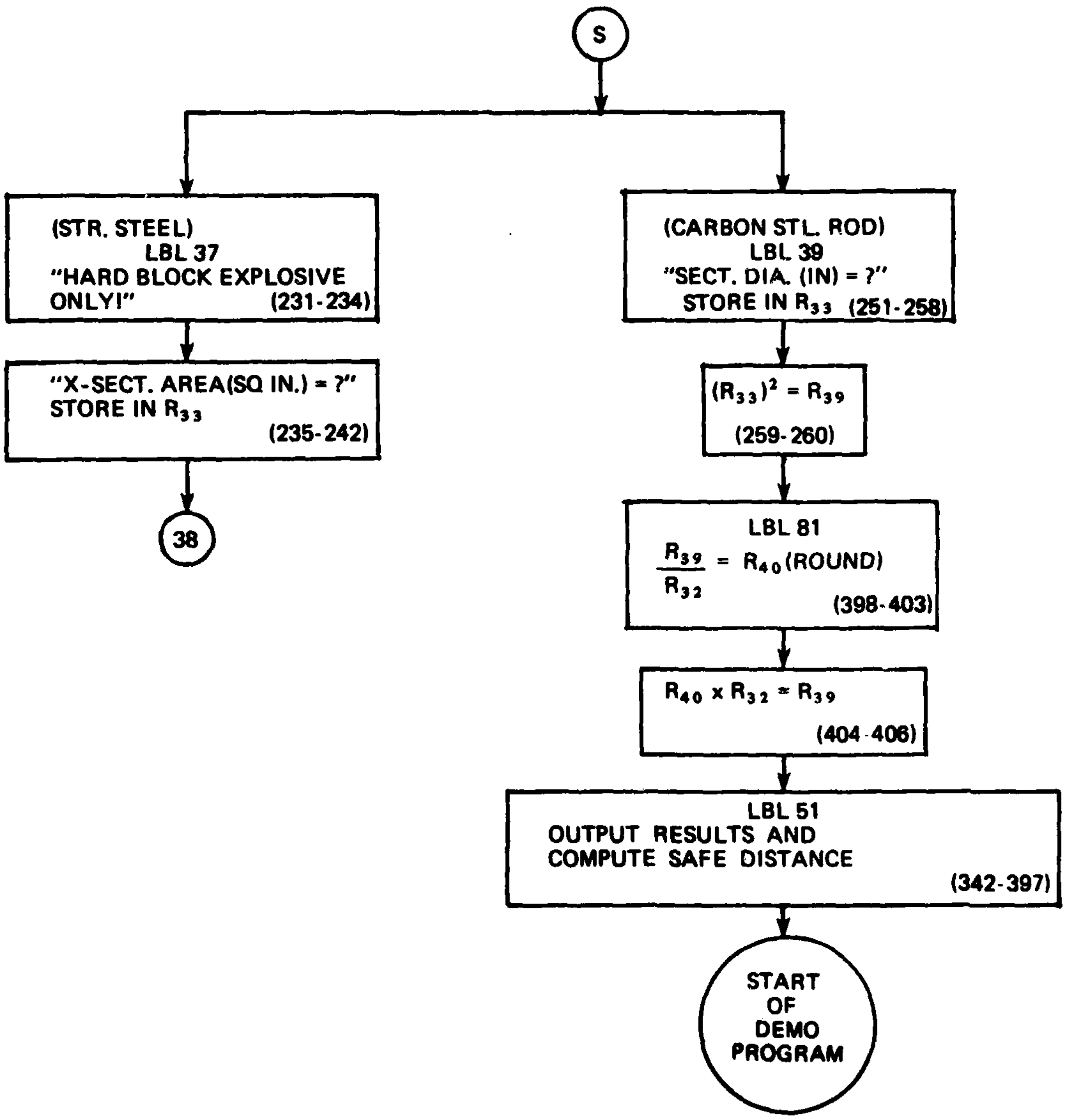


Figure D4. (Cont'd).

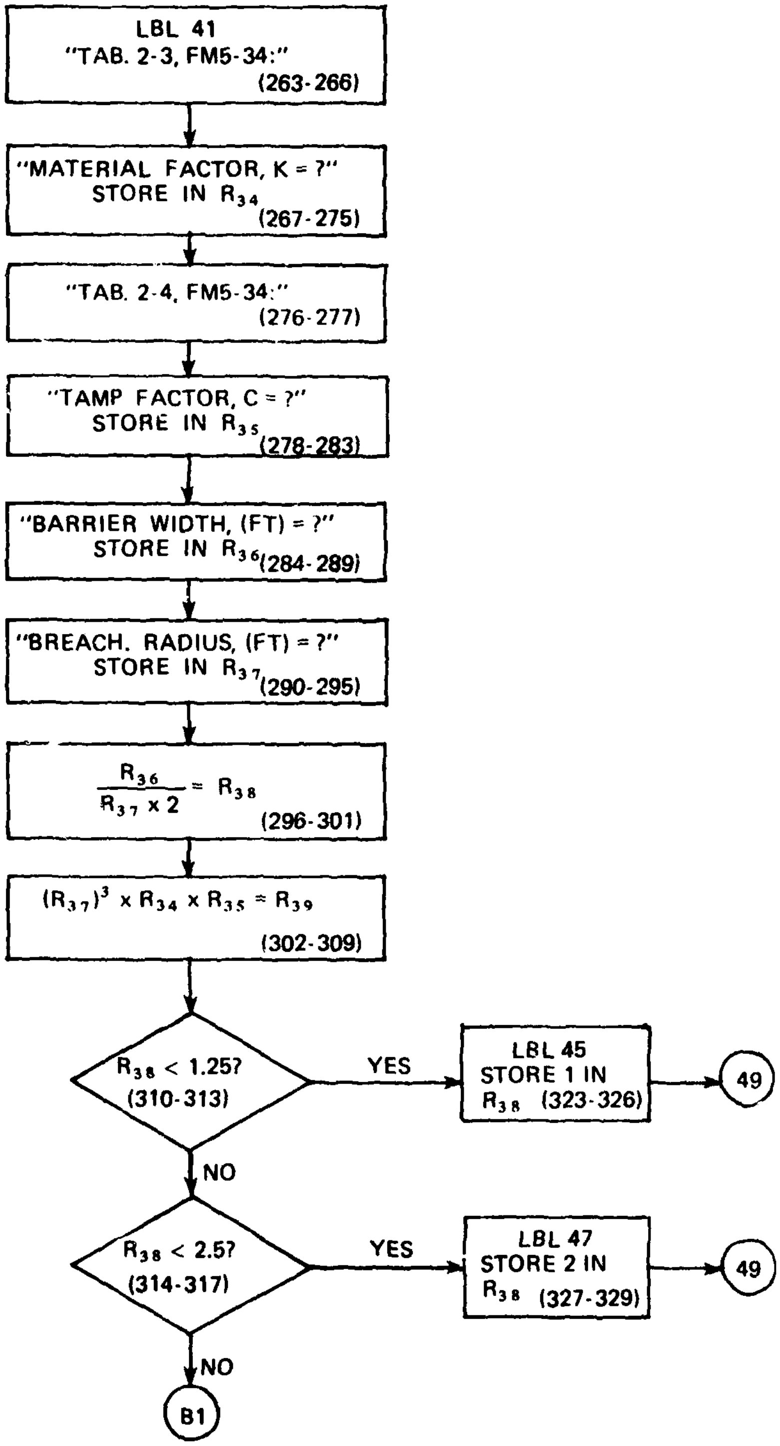


Figure D4. (Cont'd).

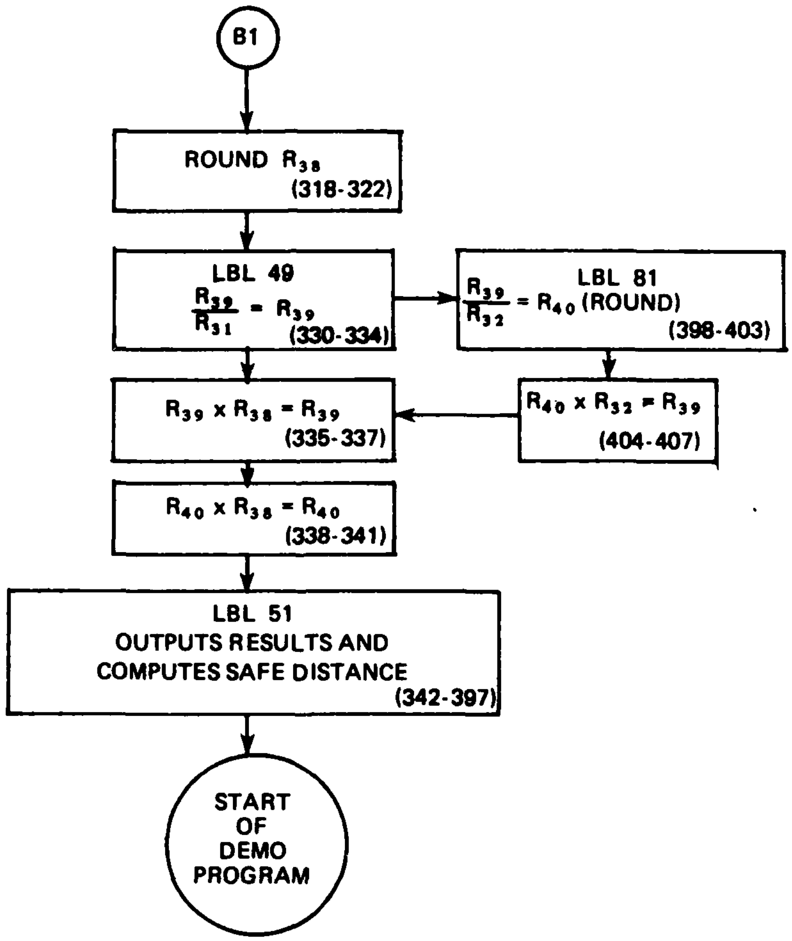


Figure D4. (Cont'd).

```
01♦LBL "DEMO"
02 41
03 XEQ "*S"
04 CF 00

05♦LBL 01
06 30.04
07 XEQ "*C"
08 FIX 1

09♦LBL 03
10 "EXPLOSIVE TYPE:"
11 XEQ "*D"
12 "TNT"
13 XEQ "*Y"
14 FS? 10
15 GTO 05
16 "M112 C4(1.25LB)"
17 XEQ "*Y"
18 FS? 10
19 GTO 07
20 "M5A1 C4(2.5LB)"
21 XEQ "*Y"
22 FS? 10
23 GTO 09
24 "DYNAMITE,M1"
25 XEQ "*Y"
26 FS? 10
27 GTO 11
28 "TETRYTOL"
29 XEQ "*Y"
30 FS? 10
31 GTO 13
32 "M118 SHEET(0.5L"
33 "⊢B)"
34 XEQ "*Y"
35 FS? 10
36 GTO 15
37 "M186 ROLL(25LB)"
38 XEQ "*Y"
39 FS? 10
40 GTO 17
41 GTO 03

42♦LBL 05
43 1
44 STO 31
45 STO 32
46 "BLOCKS,TNT,1LB"
47 XEQ "*Y"
48 FS? 10
49 GTO 19
50 .5
51 STO 32
52 GTO 19

52♦LBL 07
54 1.34
55 STO 31
56 1.25
57 STO 32
58 GTO 19

59♦LBL 09
60 1.34
61 STO 31
62 2.5
63 STO 32
64 GTO 19

65♦LBL 11
66 .92
67 STO 31
68 .5
69 STO 32
70 GTO 19

71♦LBL 13
72 1.2
73 STO 31
74 2.5
75 STO 32
76 GTO 19

77♦LBL 15
78 1.14
79 STO 31
80 .5
81 STO 32
82 GTO 19

83♦LBL 17
84 1.14
85 STO 31
86 25
87 STO 32

88♦LBL 19
89 "APPLICATION:"
90 XEQ "*D"
91 "CUT TIMBER"
92 XEQ "*Y"
93 FS? 10
94 GTO 21
95 "CUT STEEL"
96 XEQ "*Y"
97 FS? 10
98 GTO 31
99 "BREACH"
100 XEQ "*Y"
101 FS? 10
102 GTO 41
103 GTO 19

104♦LBL 21
105 33
106 STO 24
107 "TIMBER DIA.(IN)"
108 180
109 ENTER↑
110 .5
111 XEQ "*I"

112♦LBL 23
113 "CHARGE PLACEMEN"
114 "⊢T:"
115 XEQ "*D"
116 "ABATIS"
117 XEQ "*Y"
118 FS? 10
119 GTO 25
120 "EXTERN."
121 XEQ "*Y"
122 FS? 10
123 GTO 27
124 "INTERN."
125 XEQ "*Y"
126 FS? 10
127 GTO 29
128 GTO 23

129♦LBL 25
130 RCL 33
131 X↑2
132 50
133 /
134 RCL 31
135 /
136 XEQ 81
137 GTO 51
```

Figure D5. "DEMO" program listing.

```
138♦LBL 27
139 RCL 33
140 X↑2
141 40
142 /
143 RCL 31
144 /
145 XEQ 81
146 GTO 51

147♦LBL 29
148 RCL 33
149 X↑2
150 250
151 /
152 RCL 31
153 /
154 XEQ 81
155 GTO 51

156♦LBL 31
157 "TYPE STEEL:"
158 XEQ "*D"
159 "RR.RAIL"
160 XEQ "*Y"
161 FS? 10
162 GTO 32
163 "SHAPE, ROUND"
164 XEQ "*Y"
165 FS? 10
166 GTO 34
167 "STR.STL."
168 XEQ "*Y"
169 FS? 10
170 GTO 37
171 "CARBON STL.ROD"
172 XEQ "*Y"
173 FS? 10
174 GTO 39
175 GTO 31

176♦LBL 32
177 33
178 STO 24
179 "RAIL.HT.(IN)"
180 9
181 ENTER↑
182 1
183 XEQ "*I"
184 "RR.FROG"
185 XEQ "*Y"
186 FS? 10
187 GTO 36
188 5
189 RCL 33
190 X<Y?
191 GTO 33
192 1
193 XEQ 81
194 GTO 51

195♦LBL 33
196 .5
197 XEQ 81
198 GTO 51

199♦LBL 34
200 33
201 STO 24
202 "BAR.DIA.(IN)"
203 24
204 ENTER↑
205 0
206 XEQ "*I"
207 1
208 RCL 33
209 X<=Y?
210 GTO 35
211 2
212 RCL 33
213 X<Y?
214 GTO 36
215 RDL 33
216 2
217 /
218 X↑2
219 PI
220 *
221 STO 33
222 GTO 38

223♦LBL 35
224 1
225 XEQ 81
226 GTO 51

227♦LBL 36
228 2
229 XEQ 81
230 GTO 51

231♦LBL 37
232 "HARD BLOCK EXPL"
233 "⊢O. ONLY!"
234 XEQ "*D"
235 33
236 STO 24
237 "X-SEC.AREA(SQ."
238 "⊢IN)"
239 99
240 ENTER↑
241 0
242 XEQ "*I"

243♦LBL 38
244 RCL 33
245 3
246 *
247 8
248 /
249 XEQ 81
250 GTO 51

251♦LBL 39
252 33
253 STO 24
254 "SECT.DIA.(IN)"
255 99
256 ENTER↑
257 0
258 XEQ "*I"
259 RCL 33
260 X↑2
261 XEQ 81
262 GTO 51

263♦LBL 41
264 SF 00
265 "TAB.2-3,FM-34:"
266 XEQ "*D"
267 34
268 STO 24
269 FIX 2
270 "MATERIAL FACTOR"
271 "⊢,K"
272 1.76
273 ENTER↑
274 .07
275 XEQ "*I"
276 "TAB.2-4,FM5-34:"
277 XEQ "*D"
278 FIX 1
```

Figure D5. (Cont'd).

```
279 "TAMP FACTOR,C"
280 3.6
281 ENTER↑
282 1
283 XEQ "*I"
284 "BARRIER WIDTH,("
285 "⊢FT)"
286 999
287 ENTER↑
288 0
289 XEQ "*I"
290 "BREACH. RADIUS,"
291 "⊢(FT)"
292 99
293 ENTER↑
294 0.1
295 XEQ "*I"
296 RCL 36
297 RCL 37
298 /
299 2
300 /
301 STO 38
302 RCL 37
303 3
304 Y↑X
305 RCL 34
306 *
307 RCL 35
308 *
309 STO 39
310 1.25
311 RCL 38
312 X<Y?
313 GTO 45
314 2.5
315 RCL 38
316 X<Y?
317 GTO 47
318 RCL 38
319 FIX 0
320 RND
321 STO 38
322 GTO 49

323♦LBL 45
324 1
325 STO 38
326 GTO 49

327♦LBL 47
328 2
329 STO 38

330♦LBL 49
331 RCL 39
332 RCL 31
333 /
334 XEQ 81
335 RCL 38
336 *
337 STO 39
338 RCL 40
339 RCL 38
340 *
341 STO 40

342♦LBL 51
343 ADV
344 FIX 1
345 "REQD.EXPLO,LBS"
346 RCL 39
347 XEQ "*O"
348 FIX 0
349 "#EXPLO.UNITS"
350 RCL 40
351 XEQ "*O"
352 RCL 39
353 5
354 X>Y?
355 XEQ 52
356 FC? 00
357 GTO 53
358 "#CHARGES"
359 RCL 38
360 XEQ "*O"
361 GTO 53

362♦LBL 52
363 FC? 00
364 RTN
365 "REINF.CONCR:USE"
366 "⊢5LB.MIN."
367 XEQ "*D"
368 "DENSE CONCR:USE"
369 "⊢3LB.MIN."
370 XEQ "*D"
371 RTN

372♦LBL 53
373 27
374 RCL 39
375 X<=Y?
376 GTO 55
377 1
378 ENTER↑
379 3
380 /
381 Y↑X
382 100
383 *
384 STO 30
385 GTO 57

386♦LBL 55
387 300
388 STO 30

389♦LBL 57
390 "IN OPEN,SAFE DI"
391 "⊢ST,M"
392 RCL 30
393 XEQ "*O"
394 XEQ "*P"
395 CF 00
396 STOP
397 GTO 01

398♦LBL 81
399 STO 39
400 RCL 32
401 /
402 XEQ "*R"
403 STO 40
404 RCL 32
405 *
406 STO 39
407 RTN
408 "2/2/82"
409 .END.
```

Figure D5. (Cont'd).

Table D1

"DEMO" Program Example -- Without Printer

Step	Press	Resulting Display
1	XEQ	
	ALPHA	
	DEMO	
	ALPHA	EXPLOSIVE TYPE:
2	R/S	TNT (Y/N)?
3	N R/S	M112 C4 (1.25 LB) (Y/N)?
4	N R/S	M5A1 C4 (2.5 LB) (Y/N)?
5	N R/S	DYNAMITE, M1 (Y/N)?
6	Y R/S	APPLICATION:
7	R/S	CUT TIMBER (Y/N)?
8	N R/S	CUT STEEL (Y/N)?
9	N R/S	BREACH (Y/N)?
10	Y R/S	TAB. 2-3, FM 5-34:
11	R/S	MATERIAL FACTOR, K = ?
12	.54 R/S	TAB. 2-4, FM 5-34:
13	R/S	TAMP FACTOR, C = ?
14	2 R/S	BARRIER WIDTH, (FT) = ?
15	42 R/S	BREACH. RADIUS, (FT) = ?
16	7.5 R/S	REQD. EXPLO, LBS = 1,486.5
17	R/S	# EXPLO. UNITS = 2,973.
18	R/S	# CHARGES = 3.
19	R/S	OPEN, SAFE DIST, M = 1,141.
20	R/S	END PROGRAM

Table D2

"DEMO" Program -- Special Regiscer Uses

Register Number	Register Contents
30	"In open" safe distance in meters
31	Relative effectiveness factor (RE) of specified explosive
32	Explosive unit weights
33	Timber diameter in inches (or) rail height in inches (or) bar diameter in inches (or) cross-sectional area in square inches (or) sectional diameter in inches
34	Material factor, K
35	Tamp factor, C
36	Barrier width in feet
37	Breaching radius in feet
38	Number of charges required
39	Pounds of explosive required
40	Number of explosive units required

Table D3

"DEMO" Program -- Functions, By Label

Label	Purpose
DEMO	Marks beginning of program; checks size allocation
01	Clears register
03	Presents explosive-type menu
05	Loads TNT effectiveness factor (EF) in R_{31} and issue size into R_{32}
07	Loads M112 C4 (1.25 lb) effectiveness factor (EF) in R_{31} and issue size into R_{32}
09	Loads M5A1 C4 (2.5 lb) effectiveness factor (EF) in R_{31} and issue size into R_{32}
11	Loads dynamite, M1 effectiveness factor (EF) in R_{31} and issue size into R_{32}
13	Loads Tetrytol effectiveness factor (EF) in R_{31} and issue size into R_{32}
15	Loads M118 sheet (0.5 lb) effectiveness factor (EF) in R_{31} and issue size into R_{32}
17	Loads M186 roll (25 lb) effectiveness factor (EF) in R_{31} and issue size into R_{32}
19	Presents application menu
21	Inputs timber diameter in inches
23	Presents charge placement menu (for cut-timber application)
25	Computes values for abatis charge placement
27	Computes values for external charge placement
29	Computes values for internal charge placement
31	Presents menu for cut steel applications
32	Inputs rail height; determines if railroad frogs are used; creates/stores values for railroad rails > 5 in.
33	Creates/stores values for railroad rails < 5 in.

Table D3 (Cont'd)

Label	Purpose
34	Inputs bar diameter; computes/stores values for mild steel round shape rods
35	Creates/stores values for mild steel round shape rods with diameters $\leq$ 1 in.
36	Creates/stores values for mild steel round shape rods with diameters between 1 and 2 in. and RR frogs
37	Inputs cross-sectional area in square inches for structural steel sections
38	Computes values for structural steel sections
39	Inputs diameter of section; computes values for high carbon steel rods
41	Inputs K-factor, C-factor, barrier width, and breaching radius; computes values for breach wall application
45	Intermediate calculation for breach wall charges
47	Intermediate calculation for breach wall charges
49	Intermediate calculation for breach wall charges
51	Outputs values for all applications
52	Displays minimum limitations for breach wall
53	Computes safe distance
55	Intermediate computation for safe distance
57	Outputs safe distance; displays program end
Local Sub-Routine	
81	Computes number of units and pounds of explosive

APPENDIX E:
"MINES" PROGRAM DETAILS

This appendix provides detailed information about the "MINES" program. Figure E1 shows the typical sequence of events and the options that a user encounters when executing this program. Table E1 is an example of the specific steps that must be followed to solve a problem when one uses the HP-41 calculator without a printer attached. In this example, the program determines the logistical requirements for a minefield 400-m long and 400-m deep, with an antitank, antipersonnel fragmentation, antipersonnel blast (AT-APF-APB) mine density of 1-1-0 and an AT-APF-APB mine irregular outer edge (IOE) cluster composition of 1-2-2.

In step 2 of Table E1, the user has to press the R/S key to restart the program after only part of the message has been displayed. If a printer were attached and set to the "NORM" printer mode, the program would advance automatically, and steps 13 through 31 would be executed automatically. Figure E2 shows two examples of using the "MINES" program with a printer attached.

Abbreviations used in "MINES" are:

Symbol	Meaning
APB	Antipersonnel Blast
APF	Antipersonnel Fragmentation
AT	Antitank
IOE	Irregular Outer Edge
M	Meter(s)
MAX	Maximum
MMF	Main Minefield
RL	Reel(s)
(Y/N)	Yes/No
#	Number

Eight sets of operating limits for input variables are in the program:

Variable	Units	Minimum	Maximum
Mine densities:			
AT mines/m	Mines/meter	0	4
APF mines/m	Mines/meter	0	16
APB mines/m	Mines/meter	0	16
IOE cluster composition:			
No. of AT mines	Each	0	1
No. of APF mines	Each	0	5
No. of APB mines	Each	0	5
Minefield length	Meters	0	5000
Minefield depth	Meters	0	999

The maximum values for the IOE cluster composition were determined by two rules: the maximum number of antitank mines per cluster is always one; the maximum total number of mines in any given cluster is five. Algorithms used in the program were taken from FM 5-34 (pp 54-59), and from FM 20-32.[3] Critical assumptions and formulas are as follows:

1. Number of IOE live clusters = $\frac{\text{meters of front}}{9}$.

2. Number of mines in main minefield = meters of front x desired density.

3. Number of mines in IOE = number of IOE live clusters x densities for IOE representative cluster.

4. Total number of mines = (number of mines in MMF + number of mines in IOE) x 110 percent.

5. Number of strips required — the larger value of:

   a. (3/5) x (total mine density), or

   b. 3 x (AT density).

6. Number of sandbags = (9S + 3) x (number of IOE clusters), where S = number of lettered strips.

7. Number of pickets and signs = $\frac{\text{meters of barbed wire}}{30}$.

8. Number of man-hours (MN-HR):

$$\left[\frac{\text{Total AT Mines}}{\text{4 AT Mines/MN-HR}} + \frac{\text{Total APF Mines}}{\text{8 APF Mines/MN-HR}} + \frac{\text{Total APB Mines}}{\text{16 APB Mines/MN-HR}}\right] \times 1.2.$$

9. Man-hour adjustment: increase man-hours by 50 percent if work is to be done at night.

10. Number of barbed wire reels = $\frac{[4(\text{depth} + \text{width}) + 320] \times (1.4)}{400}$.

"MINES" uses registers 30 through 52 to store the values described in Table E2. The program sets general purpose flag "01" if work is to be done at night.

Table E3 describes the general function of each part of the program, by label. Figure E3, a label wiring diagram, shows how the different parts of the program relate to each other. A circular loop on the diagram indicates a

---

[3] Mine/Countermine Operations at the Company Level, FM 20-32 (HQ, DA, November 1976).

return to the same label. A two-headed arrow pointing to and from another label indicates the program executes that part as a local subroutine, then returns to the main program. Global subroutines, used by all the major application programs on the MILENG1/UTIL module, are not shown on the wiring diagram, but are described separately in Appendix G.

Figure E4 presents a detailed flowchart of the "MINES" program, and Figure E5 lists the program steps.

User executes "MINES"

User enters mine densities

User enters IOE cluster composition

User indicates if night work is to be done

User inputs minefield length in meters

User inputs minefield depth in meters

Program computes logistical requirements

User retrieves results

Figure E1. "MINES" program sequence of events.

```
              XEQ "MINES"
ENTER MINE DENSITY:
#AT/M=?
              1.00       RUN
#APF/M=?
              1.00       RUN
#APB/M=?
              2.00       RUN
IOE CLUSTER COMPOSITION:
#AT=?
                1.       RUN
#APF=?
                1.       RUN
#APB=?
                1.       RUN
DO AT NIGHT(Y/N)?
Y                        RUN
FIELD LENGTH,(M)=?
              400.       RUN
FIELD DEPTH,(M)=?
              100.       RUN

TOTAL MINES:
#AT=490.
#APF=490.
#APB=930.
IOE MINES:
#AT=45.
#APF=45.
#APB=45.
MMF MINES:
#AT=400.
#APF=400.
#APB=800.
#IOE CLUSTERS=45.
#STRIPS=3.
2-STRAND,4-SIDE FENCE:
#WIRE(RL)=9.
#SIGNS,PICKETS=109.
#SANDBAGS=1,350.
MANHOURS=436.
END PROGRAM
```

```
              XEQ "MINES"
ENTER MINE DENSITY:
#AT/M=?
              3.00       RUN
#APF/M=?
              4.00       RUN
#APB/M=?
              8.00       RUN
IOE CLUSTER COMPOSITION:
#AT=?
                1.       RUN
#APF=?
                2.       RUN
#APB=?
                2.       RUN
DO AT NIGHT(Y/N)?
N                        RUN
FIELD LENGTH,(M)=?
              400.       RUN
FIELD DEPTH,(M)=?
              400.       RUN

TOTAL MINES:
#AT=1,370.
#APF=1,859.
#APB=3,619.
IOE MINES:
#AT=45.
#APF=90.
#APB=90.
MMF MINES:
#AT=1,200.
#APF=1,600.
#APB=3,200.
#IOE CLUSTERS=45.
#STRIPS=9.
2-STRAND,4-SIDE FENCE:
#WIRE(RL)=13.
#SIGNS,PICKETS=165.
#SANDBAGS=3,780.
MANHOURS=962.
END PROGRAM
```

Figure E2. "MINES" program examples -- with printer.

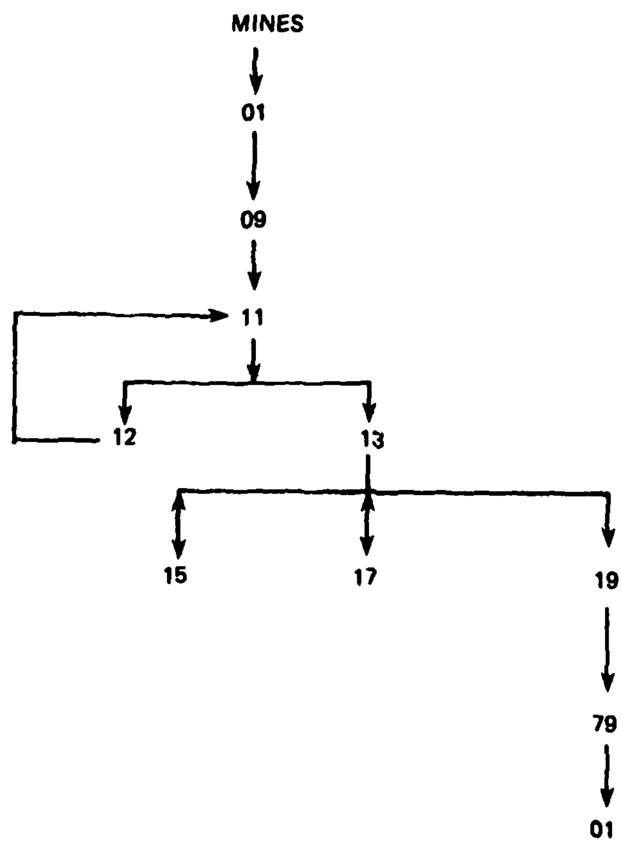


Figure E3. "MINES" program label wiring diagram.

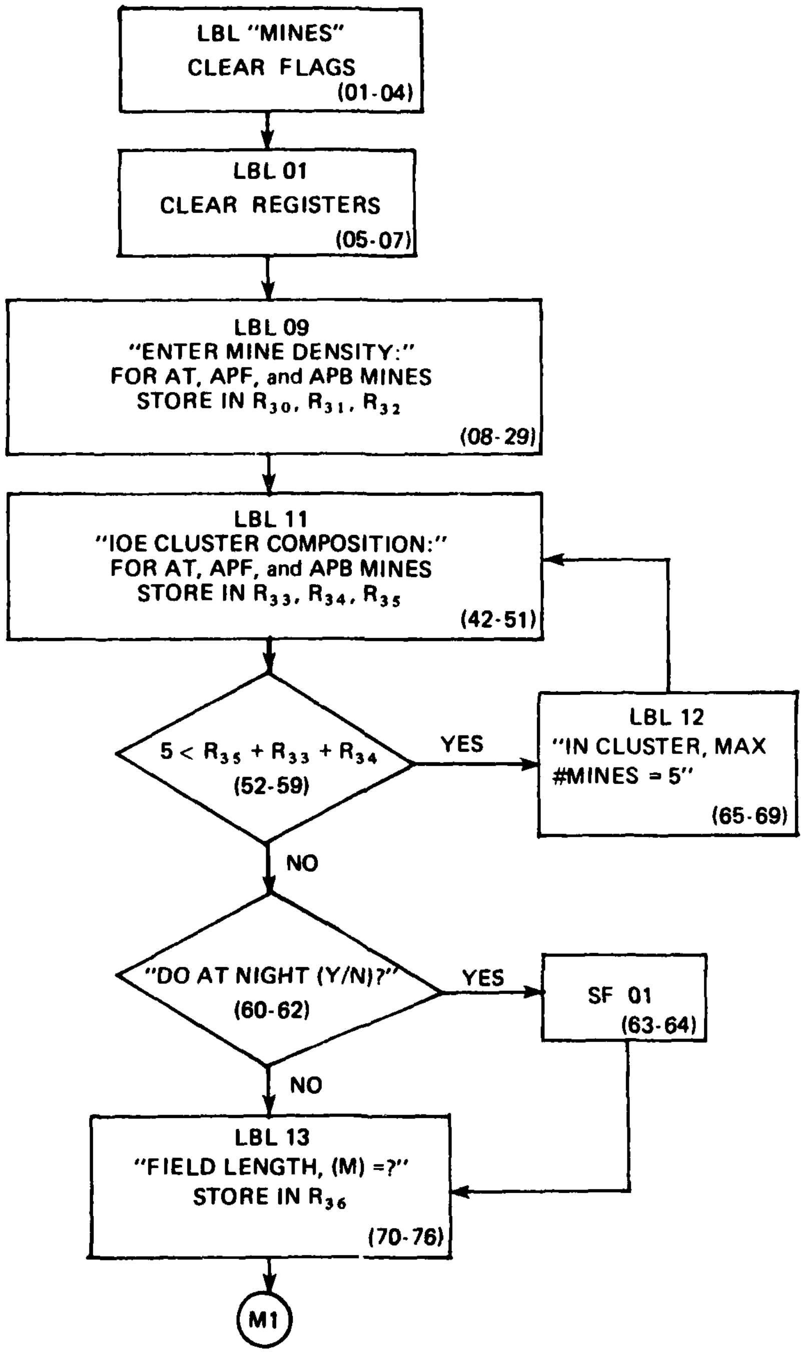


Figure E4. "MINES" program -- detailed flowchart.

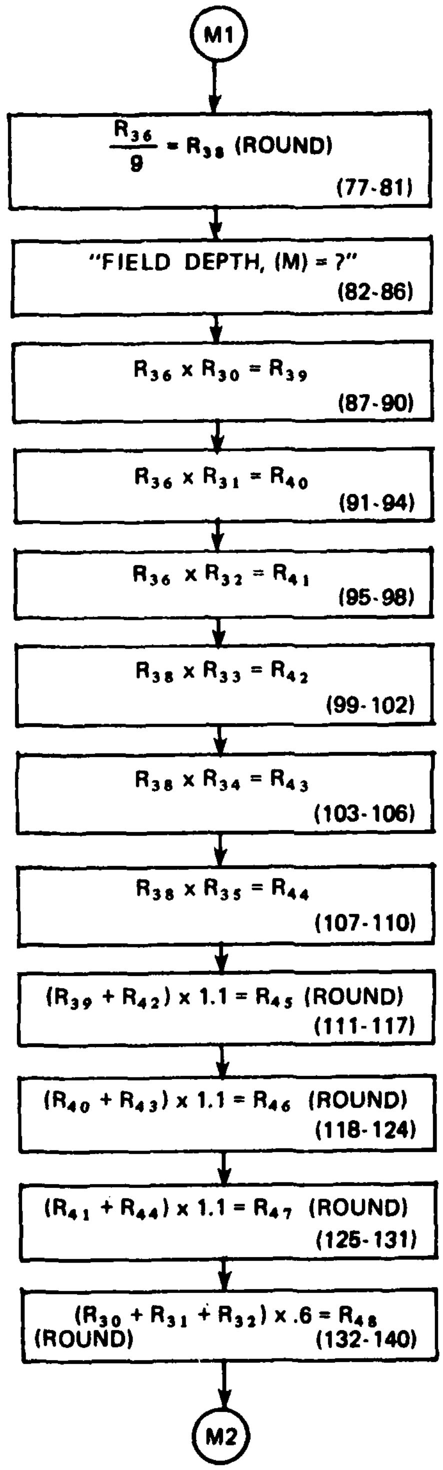


Figure E4. (Cont'd).

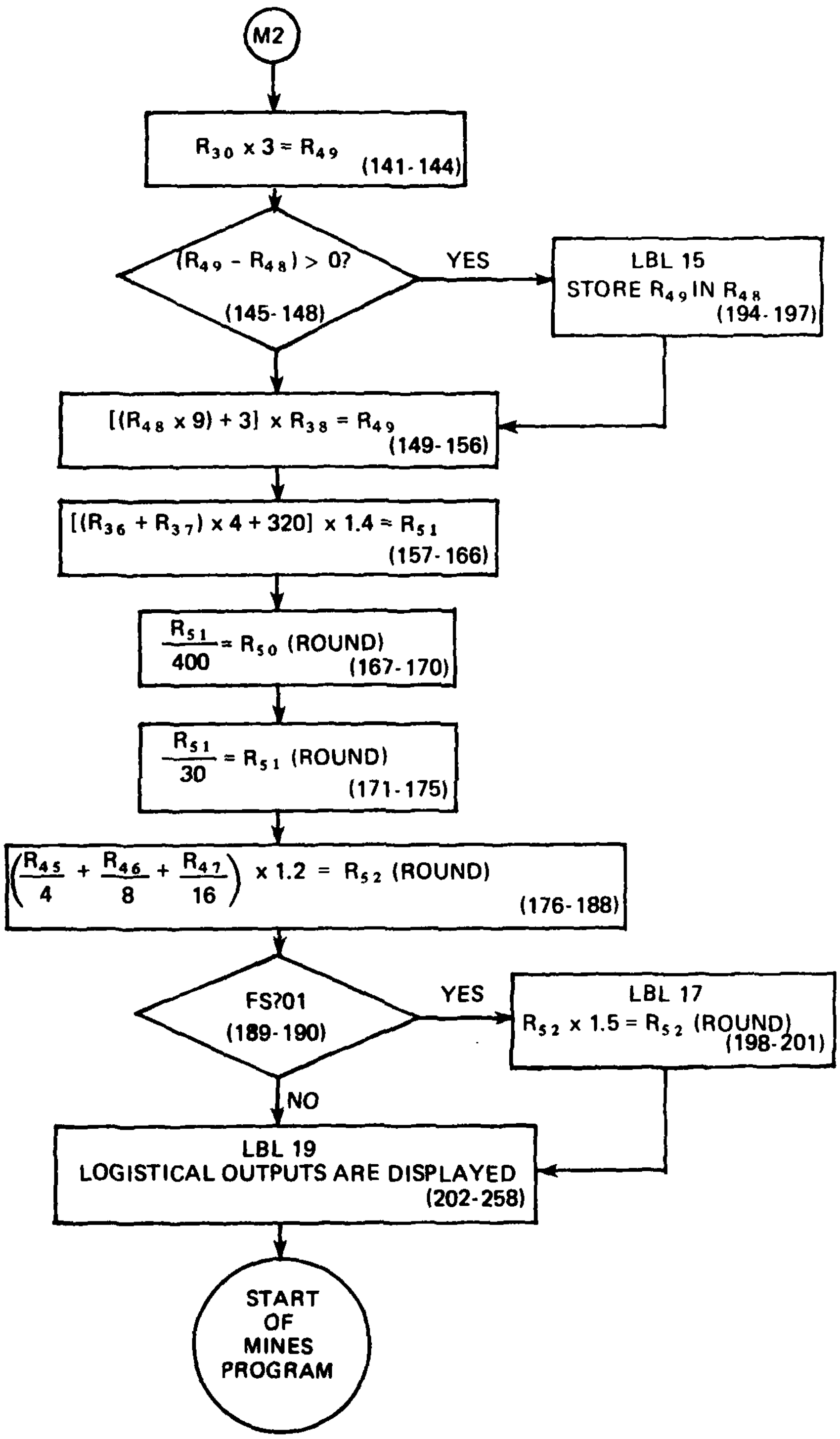


Figure E4. (Cont'd).

```
01♦LBL "MINES"
02 53
03 XEQ "*S"
04 XEQ "*F"

05♦LBL 01
06 30.052
07 XEQ "*C"

08♦LBL 09
09 "ENTER MINE DENS"
10 "⊢ITY:"
11 XEQ "*D"
12 30
13 STO 24
14 FIX 2
15 "#AT/M"
16 4
17 ENTER↑
18 0
19 XEQ "*I"
20 "#APF/M"
21 16
22 ENTER↑
23 0
24 XEQ "*I"
25 "#APB/M"
26 16
27 ENTER↑
28 0
29 XEQ "*I"

30♦LBL 11
31 "IOE CLUSTER COM"
32 "⊢POSITION:"
33 XEQ "*D"
34 33
35 STO 24
36 FIX 0
37 "#AT"
38 1
39 ENTER↑
40 0
41 XEQ "*I"
42 "#APF"
43 5
44 ENTER↑
45 0
46 XEQ "*I"
47 "#APB"
48 5
49 ENTER↑
50 0
51 XEQ "*I"
52 RCL 35
53 RCL 33
54 +
55 RCL 34
56 +
57 5
58 X<Y?
59 GTO 12
60 "DO AT NIGHT"
61 XEQ "*Y"
62 FS? 10
63 SF 01
64 GTO 13

65♦LBL 12
66 "IN CLUSTER, MAX"
67 "⊢ #MINES=5:"
68 XEQ "*D"
69 GTO 11

70♦LBL 13
71 "FIELD LENGTH,(M"
72 "⊢)"
73 5000
74 ENTER↑
75 0
76 XEQ "*I"
77 RCL 36
78 9
79 /
80 XEQ "*R"
81 STO 38
82 "FIELD DEPTH,(M)"
83 999
84 ENTER↑
85 0
86 XEQ "*I"
87 RCL 36
88 RCL 30
89 *
90 STO 39
91 RCL 36
92 RCL 31
93 *
94 STO 40
95 RCL 36
96 RCL 32
97 *
98 STO 41
99 RCL 38
100 RCL 33
101 *
102 STO 42
103 RCL 38
104 RCL 34
105 *
106 STO 43
107 RCL 38
108 RCL 35
109 *
110 STO 44
111 RCL 39
112 RCL 42
113 +
114 1.1
115 *
116 XEQ "*R"
117 STO 45
118 RCL 40
119 RCL 43
120 +
121 1.1
122 *
123 XEQ "*R"
124 STO 46
125 RCL 41
126 RCL 44
127 +
128 1.1
129 *
130 XEQ "*R"
131 STO 47
132 RCL 30
133 RCL 31
134 +
135 RCL 32
136 +
137 .6
138 *
139 XEQ "*R"
140 STO 48
141 RCL 30
142 3
143 *
144 STO 49
145 RCL 48
146 -
147 X>0?
148 XEQ 15
```

Figure E5. "MINES" program listing.

```
149 RCL 48
150 9
151 *
152 3
153 +
154 RCL 38
155 *
156 STO 49
157 RCL 36
158 RCL 37
159 +
160 4
161 *
162 320
163 +
164 1.4
165 *
166 STO 51
167 400
168 /
169 XEQ "*R"
170 STO 50
171 RCL 51
172 30
173 /
174 XEQ "*R"
175 STO 51
176 RCL 45
177 4
178 /
179 RCL 46
180 8
181 /
182 +
183 RCL 47
184 16
185 /
186 +
187 1.2
188 *
189 FS? 01
190 XEQ 17
191 XEQ "*R"
192 STO 52
193 GTO 19

194♦LBL 15
195 RCL 49
196 STO 48
197 RTN

198♦LBL 17
199 1.5
200 *
201 RTN

202♦LBL 19
203 ADV
204 "TOTAL MINES:"
205 XEQ "*D"
206 "#AT"
207 RCL 45
208 XEQ "*O"
209 "#APF"
210 RCL 46
211 XEQ "*O"
212 "#APB"
213 RCL 47
214 XEQ "*O"
215 "IOE MINES:"
216 XEQ "*D"
217 "#AT"
218 RCL 42
219 XEQ "*O"
220 "#APF"
221 RCL 43
222 XEQ "*O"
223 "#APB"
224 RCL 44
225 XEQ "*O"
226 "MMF MINES:"
227 XEQ "*D"
228 "#AT"
229 RCL 39
230 XEQ "*O"
231 "#APF"
232 RCL 40
233 XEQ "*O"
234 "#APB"
235 RCL 41
236 XEQ "*O"
237 "#IOE CLUSTERS"
238 RCL 38
239 XEQ "*O"
240 "#STRIPS"
241 RCL 48
242 XEQ "*O"
243 "2-STRAND,4-SIDE"
244 "⊢ FENCE:"
245 XEQ "*D"
246 "#WIRE(RL)"
247 RCL 50
248 XEQ "*O"
249 "#SIGNS,PICKETS"
250 RCL 51
251 XEQ "*O"
252 "#SANDBAGS"
253 RCL 49
254 XEQ "*O"
255 "MANHOURS"
256 RCL 52
257 XEQ "*O"
258 GTO 79

259♦LBL 79
260 XEQ "*F"
261 XEQ "*P"
262 STOP
263 GTO 01
264 "2/2/82"
265 .END.
```

Figure E5. (Cont'd).

Table E1

"MINES" Program Example -- Without Printer

Step	Press	Resulting Display
1	XEQ ALPHA MINES ALPHA	ENTER MINE DENSITY:
2	R/S	# AT/M = ?
3	3 R/S	# APF/M = ?
4	4 R/S	# APB/M = ?
5	8 R/S	IOE CLUSTER COMPOSITION:
6	R/S	# AT = ?
7	1 R/S	# APF = ?
8	2 R/S	# APB = ?
9	2 R/S	DO AT NIGHT (Y/N)?
10	N R/S	FIELD LENGTH, (M) = ?
11	400 R/S	FIELD DEPTH, (M) = ?
12	400 R/S	TOTAL MINES:
13	R/S	# AT = 1,370.
14	R/S	# APF = 1,859.
15	R/S	# APB = 3,619.
16	R/S	IOE MINES:
17	R/S	# AT = 45.
18	R/S	# APF = 90.
19	R/S	# APB = 90.
20	R/S	MMF MINES:
21	R/S	# AT = 1,200.
22	R/S	# APF = 1,600.
23	R/S	# APB = 3,200.
24	R/S	# IOE CLUSTERS = 45.
25	R/S	# STRIPS = 9.
26	R/S	2-STRAND, 4-SIDE FENCE:
27	R/S	# WIRE (RL) = 13.
28	R/S	# SIGNS, PICKETS = 165.
29	R/S	# SANDBAGS = 3,780.
30	R/S	MANHOURS = 962.
31	R/S	END PROGRAM

Table E2

"MINES" Program -- Special Register Uses

Register Number	Register Contents
30	Antitank (AT) mine density
31	Antipersonnel fragmentation (APF) mine density
32	Antipersonnel blast (APB) mine density
33	AT IOE representative cluster value
34	APF IOE representative cluster value
35	APB IOE representative cluster value
36	Length of minefield front in meters
37	Depth of minefield in meters
38	Approximate number of IOE clusters
39	Number of AT in main minefield
40	Number of APF in main minefield
41	Number of APB in main minefield
42	Number of AT mines in IOE
43	Number of APF mines in IOE
44	Number of APB mines in IOE
45	Total number of AT mines in minefield
46	Total number of APF mines in minefield
47	Total number of APB mines in minefield
48	Number [(a + b + c) (0.6)] of strips required
49	Number (3a) of strips required (initial) (or) number of sandbags required (final)
50	Reels of barbed wire needed
51	Meters of barbed wire needed (initial) (or) number of signs and pickets (final)
52	Number of man-hours required

Table E3

"MINES" Program -- Functions, By Label

Label	Purpose
MINES	Marks beginning of program; checks size allocation and clear flags
01	Clears register
09	Inputs main minefield densities
11	Inputs IOE cluster composition
12	Presents error message
13	Inputs field length and depth in meters; computes logistical requirements
15	Intermediate computation
17	Creates night work factor
19	Outputs logistical requirements for a minefield
79	Displays program end and clears flags

APPENDIX F: "WIRE" PROGRAM DETAILS

This appendix provides detailed information about the "WIRE" program. Figure F1 shows the typical sequence of events and the options that a user encounters when executing the program. Table F1 is an example of the specific steps that must be followed to solve a problem when one uses the HP-41 calculator without a printer attached. The table shows the effective length, the number of 300-m sections, the amount of material, the number of man-hours, and the number of truckloads required to construct a triple standard concertina obstacle. The entanglement is for protecting an area on the forward edge of the battle area (FEBA) which has 100 m of actual front. Only one belt of wire is to be used. The construction is done at night with experienced troops.

"WIRE" presents the user with two menus -- wire use and barrier type. The order of the first menu is "tactical," "protective," then "supplemental." The second menu order is "double apron 4+2," "double apron 6+3," "high wire," "low wire," "4-wire fence," "concertina, 3-ST," and "GPBTO" (General purpose barbed tape obstacle). In step 6 of Table F1, the user has to press the R/S key to restart the program after only part of the message has been displayed. If a printer were attached and set to the "NORM" printer mode, the program would advance automatically, and steps 15 through 23 would be executed automatically. Figure F2 shows two examples of using "WIRE" program with a printer attached.

Abbreviations used in "WIRE" are:

Symbol	Meaning
DBL.	Double
EFF.LEN.	Effective Length
FEBA	Forward Edge of the Battle Area
GPBTO	General Purpose Barbed Tape Obstacle
M	Meter(s)
MED	Medium
T	Ton(s)
(Y/N)	(Yes/No)
3-STD	Triple Standard
#	Number

Five sets of operating limits for input variables are in the program:

Variable	Units	Minimum	Maximum
Camp perimeter	Meters	0	50,000
Length of front	Meters	0	50,000
Unit depth	Meters	0	50,000
Effective length	Meters	0	2,250,000
Number of belts	Each	1	9

Algorithms used in the program were taken from FM 5-34 (pp 105-109), and from FM 5-15.[4] Critical assumptions and formulas are as follows:

1. To determine the effective length for use on the FEBA:

   a. Tactical wire: (front length)x(1.25)x(number of belts)
   b. Protective wire: (front length)x(5) x (number of belts)
   c. Supplementary wire:
      (1) Forward of FEBA, (front length)x(1.25)x(number of belts)
      (2) Rear of FEBA, (unit depth)x(2.5)x(number of belts).

2. To determine the effective length for use in a base camp:

   a. Tactical wire: (mean perimeter)x(1.25)x(number of belts)
   b. Protective wire: (perimeter)x(1.10)x(number of belts)
   c. Supplementary wire: (mean perimeter)x(1.25)x(number of belts).

3. To determine material and labor for 300-m sections of various wire entanglements, requirements were taken from Table 4-3, FM 5-34.

"WIRE" uses registers 30 through 43 to store the values described in Table F2. If barbed tape, experienced troops, or driven pickets are used, or if the work is done at night, the program sets the appropriate general purpose flags "01" through "04," respectively. Flag "05" is set if triple-standard concertina or GPBTO barriers are used.

Table F3 describes the general function of each part of the program, by label. Figure F3, a label wiring diagram, shows how the different parts of the program relate to each other. A circular loop on the diagram indicates a return to the same label. A two-headed arrow pointing to and from a subroutine indicates that the program executes it as a local subroutine, then returns to the main program. Global subroutines, used by all the major application programs on the MILENG1/UTIL module, are not shown on the wiring diagram, but are described separately in Appendix G.

Figure F4 presents a detailed flowchart of the "WIRE" program, and Figure F5 lists the program steps.

---

[4] <u>Field Fortifications</u>, FM 5-15 (HQ, DA, June 1972), pp 6-10, 6-22, and 6-23.

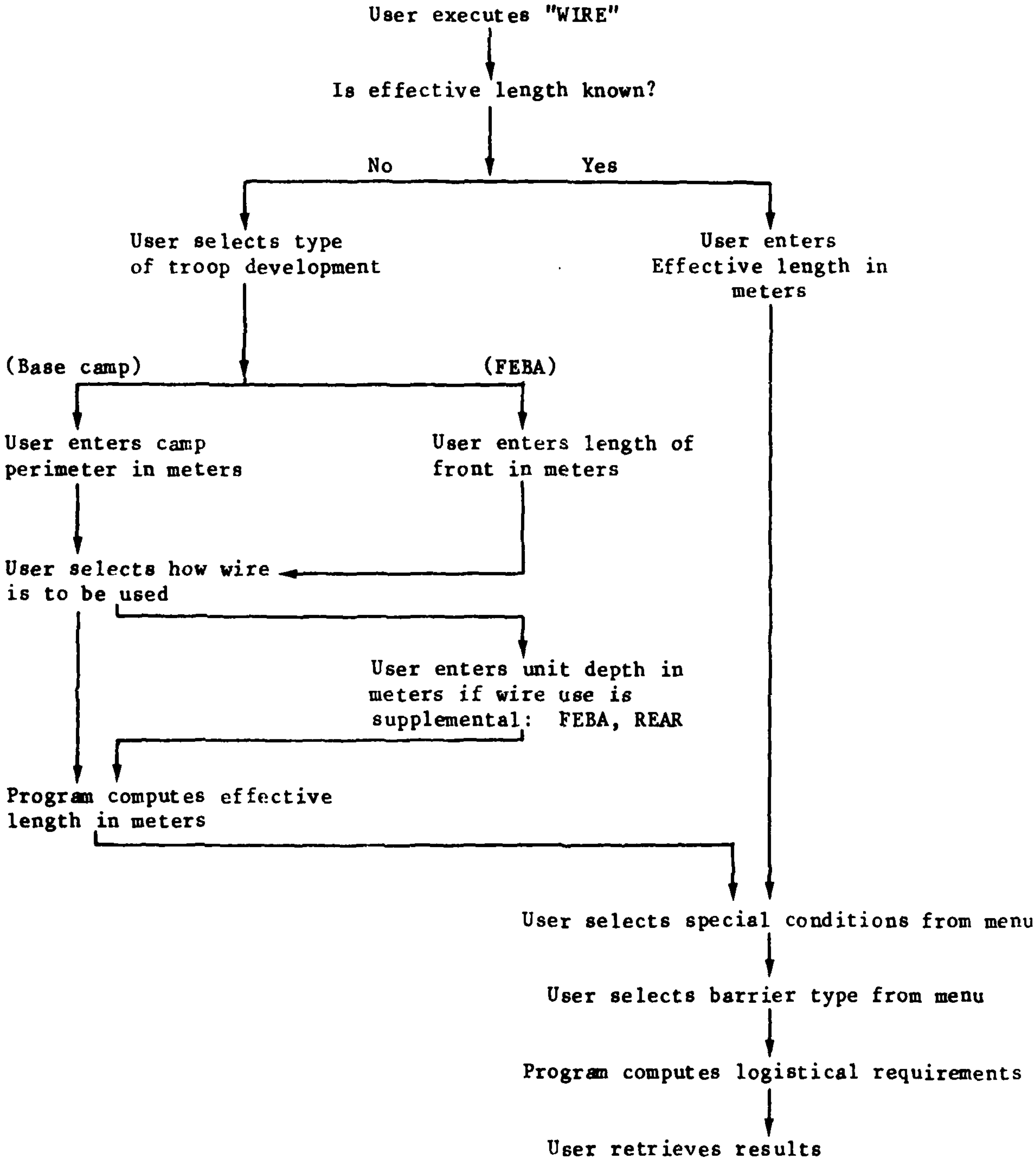


Figure F1. "WIRE" program sequence of events.

```
              XEQ "WIRE"
KNOW.EFF.LEN.(Y/N)?
Y                    RUN
EFF.LEN.(M)=?
          1,500.     RUN
USE BARBED TAPE(Y/N)?
N                    RUN
EXPERIENCED TROOPS(Y/N)?
N                    RUN
USE DRIVEN PICKETS(Y/N)?
Y                    RUN
DO AT NIGHT(Y/N)?
N                    RUN
BARRIER TYPE:
DBL APRON 4+2(Y/N)?
N                    RUN
DBL APRON 6+3(Y/N)?
N                    RUN
HIGH WIRE(Y/N)?
N                    RUN
LOW WIRE(Y/N)?
N                    RUN
4-WIRE FENCE(Y/N)?
N                    RUN
CONCERTINA,3-STD(Y/N)?
Y                    RUN

300M SECTIONS=5.0
PICKETS,LONG-800.
PICKETS,SHORT=20.
#WIRE REELS:
USING U-PICKETS=15.
USING PICKET,SCREW=15.
USING PICKET,WOOD=15.
ROLL,CONCERTINA=295.
STAPLES=1,585.
MANHOURS=150.
#2.5^TLOADS=5.4
END PROGRAM
```

```
              XEQ "WIRE"
KNOW.EFF.LEN.(Y/N)?
N                    RUN
ON,FEBA(Y/N)?
Y                    RUN
FRONT LENGTH,(M)=?
            100.     RUN
#BELTS=?
              1.     RUN
WIRE USE:
TACTICAL(Y/N)?
Y                    RUN
EFF.LEN.M=125.
USE BARBED TAPE(Y/N)?
N                    RUN
EXPERIENCED TROOPS(Y/N)?
Y                    RUN
USE DRIVEN PICKETS(Y/N)?
N                    RUN
DO AT NIGHT(Y/N)?
Y                    RUN
BARRIER TYPE:
DBL APRON 4+2(Y/N)?
Y                    RUN

300M SECTIONS=0.4
PICKETS,LONG=42.
PICKETS, SHORT=83.
#WIRE REELS:
USING U-PICKETS=6.
USING PICKET,SCREW=6.
USING PICKET,WOOD=7.
MANHOURS=25.
#2.5^TLOADS=0.3
END PROGRAM
```

Figure F2. "WIRE" program examples -- with printer.

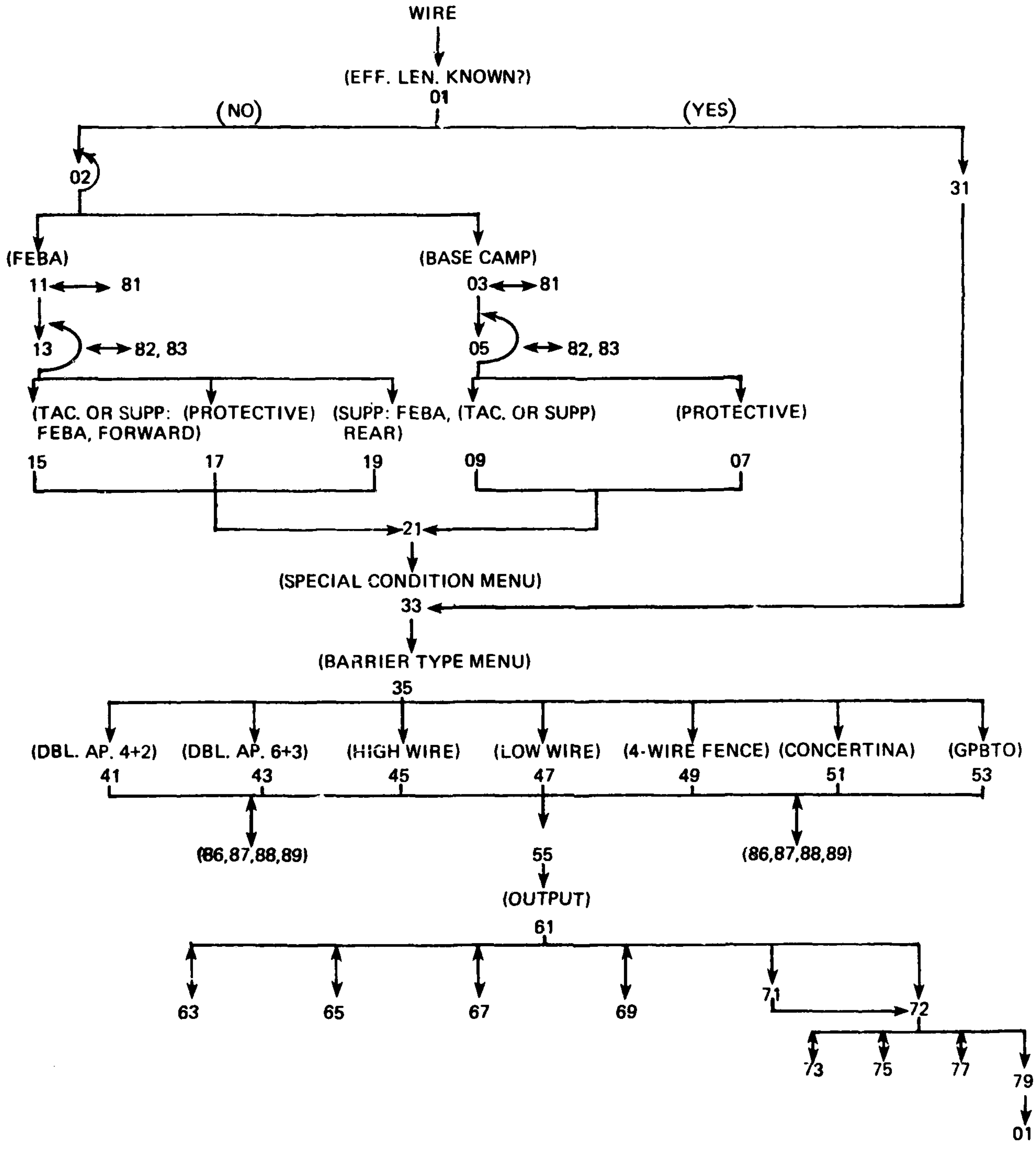


Figure F3. "WIRE" program label wiring diagram.

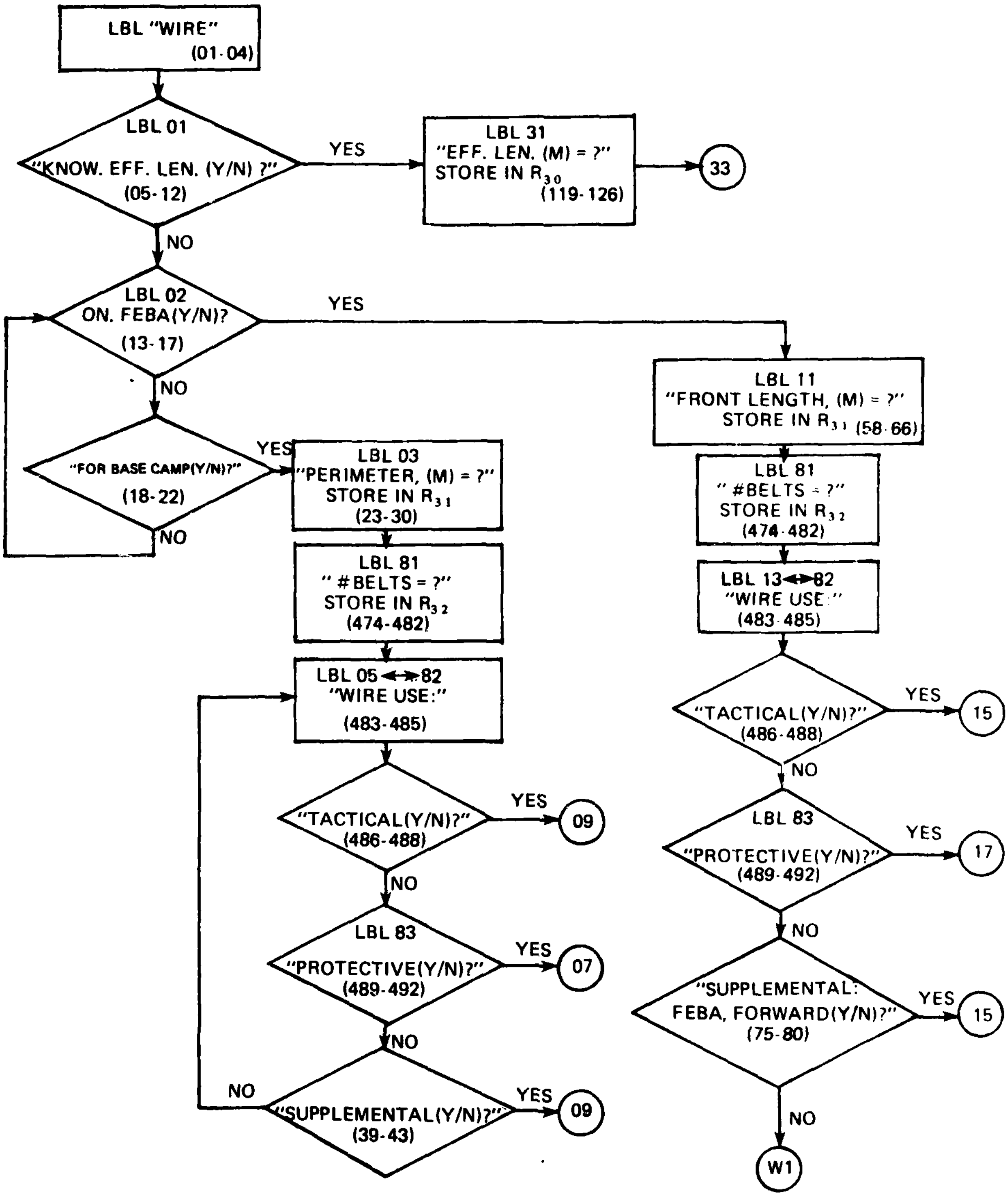


Figure F4. "WIRE" program -- detailed flowchart.

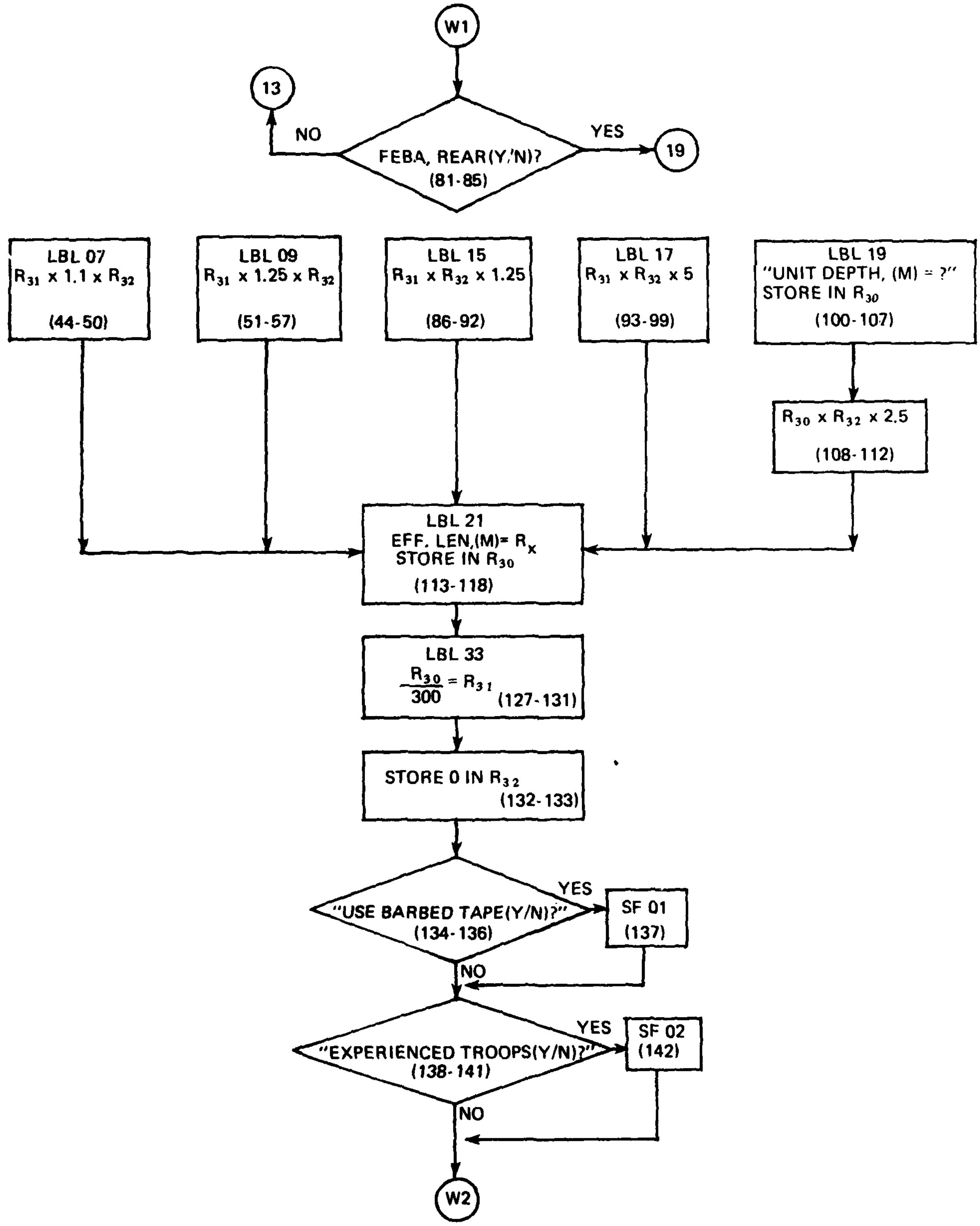


Figure F4. (Cont'd).

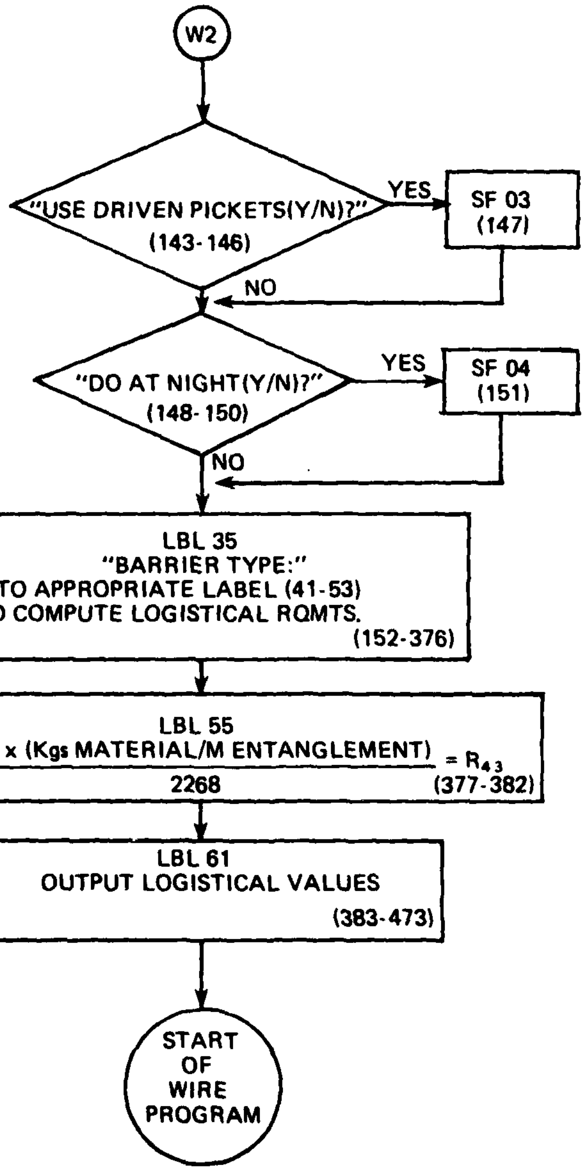


Figure F4. (Cont'd).

```
01♦LBL "WIRE"
02 44
03 XEQ "*S"
04 XEQ "*F"

05♦LBL 01
06 30.043
07 XEQ "*C"
08 FIX 0
09 "KNOW.EFF.LEN."
10 XEQ "*Y"
11 FS? 10
12 GTO 31

13♦LBL 02
14 "ON.FEBA"
15 XEQ "*Y"
16 FS? 10
17 GTO 11
18 "FOR BASE CAMP"
19 XEQ "*Y"
20 FS? 10
21 GTO 03
22 GTO 02

23♦LBL 03
24 31
25 STO 24
26 "PERIMETER,(M)"
27 50000
28 ENTER↑
29 0
30 XEQ "*I"
31 XEQ 81

32♦LBL 05
33 XEQ 82
34 FS? 10
35 GTO 09
36 XEQ 83
37 FS? 10
38 GTO 07
39 "SUPLEMENTAL"
40 XEQ "*Y"
41 FS? 10
42 GTO 09
43 GTO 05

44♦LBL 07
45 RCL 31
46 1.1
47 *
48 RCL 32
49 *
50 GTO 21

51♦LBL 09
52 RCL 31
53 1.25
54 *
55 RCL 32
56 *
57 GTO 21

58♦LBL 11
59 31
60 STO 24
61 "FRONT LENGTH,(M"
63 "⊢)"
64 50000
64 ENTER↑
65 0
66 XEQ "*I"
67 XEQ 81

68♦LBL 13
69 XEQ 82
70 FS? 10
71 GTO 15
72 XEQ 83
73 FS? 10
74 GTO 17
75 "SUPPLMENTAL:"
76 XEQ "*D"
77 "FEBA,FORWARD"
78 XEQ "*Y"
79 FS? 10
80 GTO 15
81 "FEBA,REAR"
82 XEQ "*Y"
83 FS? 10
84 GTO 19
85 GTO 13

86♦LBL 15
87 RCL 31
88 RCL 32
89 *
90 1.25
91 *
92 GTO 21

93♦LBL 17
94 RCL 31
95 RCL 32
96 *
97 5
98 *
99 GTO 21

100♦LBL 19
101 30
102 STO 24
103 "UNIT DEPTH,(M)"
104 5000
105 ENTER↑
106 0
107 XEQ "*I"
108 RCL 30
109 RCL 32
110 *
111 2.5
112 *

113♦LBL 21
114 XEQ "*R"
115 STO 30
116 "EFF.LEN,M"
117 XEQ "*O"
118 GTO 33

119♦LBL 31
120 30
121 STO 24
122 "EFF.LEN.(M)"
123 2250000
124 ENTER↑
125 0
126 XEQ "*I"
```

Figure F5. "WIRE" program listing.

```
♦LBL 33
RCL 30
300
/
STO 31
0
STO 32
"USE BARBED TAPE"
XEQ "*Y"
FS? 10
SF 01
"EXPERIENCED TRO"
"├OPS"
XEQ "*Y"
FS? 10
SF 02
"USE DRIVEN PICK"
"├ETS"
XEQ "*Y"
FS? 10
SF 03
"DO AT NIGHT"
XEQ "*Y"
FS? 10
SF 04

♦LBL 35
"BARRIER TYPE:"
XEQ "*D"
"DBL APRON 4+2"
XEQ "*Y"
FS? 10
GTO 41
"DBL APRON 6+3"
XEQ "*Y"
FS? 10
GTO 43
"HIGH WIRE"
XEQ "*Y"
FS? 10
GTO 45
"LOW WIRE"
XEQ "*Y"
FS? 10
GTO 47
"4-WIRE FENCE"
XEQ "*Y"
FS? 10
GTO 49
"CONCERTINA,3-ST"
"├D"
XEQ "*Y"
FS? 10
GTO 51
"GPBTO"
XEQ "*Y"
FS? 10
GTO 53
GTO 35

♦LBL 41
RCL 31
100
*
STO 32
2
*
STO 34
RCL 31
14
*
STO 35
RCL 31
15
*
STO 36
RCL 31
16
*
STO 37
RCL 31
19
*
STO 38
RCL 31
59
*
XEQ 86
4.6
FS? 01
3.5
GTO 55

♦LBL 43
RCL 31
66
*
STO 32
2
*
STO 34
RCL 31
13
*
STO 35
RCL 31
15
*
STO 36
RCL 31
17
*
STO 37
RCL 31
18
*
STO 38
RCL 31
49
*
XEQ 86
3.6
FS? 01
2.6
GTO 55

♦LBL 45
RCL 31
198
*
STO 32
RCL 31
17
*
STO 35
RCL 31
19
*
STO 36
RCL 31
21
*
STO 37
RCL 31
24
*
```

Figure F5. (Cont'd).

```
269 STO 38
270 RCL 31
271 79
272 *
273 XEQ 86
274 5.3
275 FS? 01
276 4
277 GTO 55

278♦LBL 47
279 RCL 31
280 100
281 *
282 STO 33
283 2
284 *
285 STO 34
286 RCL 31
287 11
288 *
289 STO 35
290 STO 36
291 STO 37
292 RCL 31
293 15
294 *
295 STO 38
296 RCL 31
297 49
298 *
299 XEQ 86
300 3.6
301 FS? 01
302 2.8
303 GTO 55

304♦LBL 49
305 RCL 31
306 100
307 *
308 STO 32
309 50
310 /
311 STO 34
312 RCL 31
313 5
314 *
315 STO 35
316 RCL 31
317 6
318 *
319 STO 36
320 RCL 31
321 7
322 *
323 STO 37
324 STO 38
325 RCL 31
326 20
327 *
328 XEQ 86
329 2.2
330 FS? 01
331 1.8
332 GTO 55

333♦LBL 51
334 RCL 31
335 160
336 *
337 STO 32
338 40
339 /
340 STO 34
341 RCL 31
342 3
343 *
344 STO 35
345 STO 36
346 STO 37
347 RCL 31
348 4
349 *
350 STO 38
351 RCL 31
352 59
353 *
354 STO 39
355 RCL 31
356 317
357 *
358 STO 40
359 RCL 31
360 30
361 *
362 SF 05
363 XEQ 86
364 8.2
365 FS? 01
366 7.3
367 GTO 55

368♦LBL 53
369 RCL 31
370 8
371 *
372 STO 41
373 RCL 31
374 SF 05
375 XEQ 86
376 2.7

377♦LBL 55
378 RCL 30
379 *
380 2268
381 /
382 STO 43

383♦LBL 61
834 ADV
385 FIX 1
386 "300M SECTIONS"
387 RCL 31
388 XEQ "*0"
389 FIX 0
390 RCL 32
391 X≠0?
392 XEQ 63
393 RCL 33
394 X≠0?
395 XEQ 65
396 RCL 34
397 X≠0?
398 XEQ 67
399 FS? 01
400 GTO 71
401 RCL 36
402 X≠0?
403 XEQ 69
404 GTO 72

405♦LBL 63
406 "PICKETS,LONG"
407 XEQ "*0"
408 RTN

409♦LBL 65
410 "PICKETS,MED."
411 XEQ "*0"
412 RTN
```

Figure F5. (Cont´d).

```
413♦LBL 67
414 "PICKETS, SHORT"
415 XEQ "*O"
416 RTN

417♦LBL 69
418 #WIRE REELS:"
419 XEQ "*D"
420 "USING U-PICKETS"
421 RCL 36
422 XEQ "*O"
423 "USING PICKET,SC"
424 "⊢REW"
425 RCL 35
426 XEQ "*O'
427 "USING PICKET,WO"
428 "⊢OD"
429 RCL 37
430 XEQ "*O"
431 RTN

432♦LBL 71
433 RCL 38
434 X=0?
435 GTO 72
436 "CASE,BARBED TAP"
437 "⊢E"
438 "XEQ "*O"

439♦LBL 72
440 RCL 39
441 X≠0?
442 XEQ 73
443 RCL 40
444 X≠0?
445 XEQ 75
446 RCL 41
447 X≠0?
448 XEQ 77
449 "MANHOURS"
450 RCL 42
451 XEQ "*O"
452 FIX 1
453 "#2.5TLOADS"
454 RCL 43
455 XEQ "*O"
456 GTO 79
457♦LBL 73
458 "ROLL,CONCERTINA"
459 XEQ "*O"
460 RTN

461♦LBL 75
462 "STAPLES"
463 XEQ "*O"
464 RTN

465♦LBL 77
466 "GPBTO UNITS"
467 XEQ "*O"
468 RTN

469♦LBL 79
470 XEQ "*P"
471 XEQ "*F"
472 STOP
473 GTO 01

474♦LBL 81
475 32
476 STO 24
477 "#BELTS"
478 9
479 ENTER↑
480 1
481 XEQ "*I"
482 RTN

483♦LBL 82
484 "WIRE USE:"
485 XEQ "*D"
486 "TACTICAL"
487 XEQ "*Y"
488 RTN

489♦LBL 83
490 "PROTECTIVE"
491 XEQ "*Y"
492 RTN

493♦LBL 86
494 STO 42
495 FS? 02
496 XEQ 87
497 FS? 03
498 XEQ 88
499 FS? 04
500 XEQ 89
501 RTN

502♦LBL 87
503 .67
504 ST* 42
505 RTN

506♦LBL 88
507 FS? 05
508 RTN
509 1.2
510 ST* 42
511 RTN

512♦LBL 89
513 1.5
514 ST* 42
515 RTN
516 "2/2/82"
517 .END.
```

Figure F5. (Cont'd).

Table F1

"WIRE" Program Example -- Without Printer

Step	Press	Resulting Display
1	XEQ	
	ALPHA	
	WIRE	
	ALPHA	KNOW. EFF. LEN. (Y/N)?
2	N R/S	ON. FEBA (Y/N)?
3	Y R/S	FRONT LENGTH, (M) = ?
4	100 R/S	# BELTS = ?
5	1 R/S	WIRE USE:
6	R/S	TACTICAL (Y/N)?
7	Y R/S	EFF. LEN, M = 125.
8	R/S	USE BARBED TAPE (Y/N)?
9	N R/S	EXPERIENCED TROOPS (Y/N)?
10	Y R/S	USE DRIVEN PICKETS (Y/N)?
11	N R/S	DO AT NIGHT (Y/N)?
12	Y R/S	BARRIER TYPE:
13	R/S	DBL APRON 4+2 (Y/N)?
14	Y R/S	300M SECTIONS = 0.4
15	R/S	PICKETS, LONG = 42.
16	R/S	PICKETS, SHORT = 83.
17	R/S	# WIRE REELS:
18	R/S	USING U-PICKETS = 6.
19	R/S	USING PICKET, SCREW = 6.
20	R/S	USING PICKET, WOOD = 7.
21	R/S	MANHOURS = 25.
22	R/S	# 2.5T LOADS = 0.3
23	R/S	END PROGRAM

Table F2

"WIRE Program -- Special Register Uses

Register Number	Register Contents
30	Unit depth in meters (initial) (or) effective length in meters (final)
31	Perimeter of camp in meters (intermediate) (or) length of front in meters (intermediate) (or) number of 300-meter sections (final)
32	Number of belts (intermediate) (or) number of long pickets needed (final)
33	Number of medium pickets needed
34	Number of short pickets needed
35	Number of barbed wire reels with screw pickets needed
36	Number of barbed wire reels with U-shaped pickets needed
37	Number of barbed wire reels with wood pickets needed
38	Number of barbed tape cases needed
39	Rolls of concertina needed
40	Number of staples needed
41	Number of GPBTO units needed
42	Number of man-hours required
43	Number of 2-1/2-ton truckloads needed

Table F3

"WIRE" Program -- Functions, By Label

Label	Purpose
Wire	Marks beginning of program; checks size allocation
01	Determines if effective length is known
02	Presents wire location menu
03	Inputs camp perimeter in meters
05	Presents wire use menu for base camp
07	Computes effective length when using protective wire for base camp defense
09	Computes effective length when using supplementary or tactical wire for base camp defense
11	Inputs length of front in meters
13	Presents wire use menu for FEBA location
15	Computes effective length when using tactical or supplementary (forward) wire for FEBA
17	Computes effective length when using protective wire for FEBA
19	Inputs unit depth; computes effective length when using supplementary (rear) wire for FEBA
21	Outputs effective length in meters
31	Inputs effective length in meters
33	Presents special conditions menu
35	Presents barrier type menu
41	Computes values for double apron, four- and two-pace entanglements
43	Computes values for double apron, six- and three-pace entanglements
45	Computes values for a high wire entanglement
47	Computes values for a low wire entanglement
49	Computes values for a four-strand cattle fence entanglement
51	Computes values for a triple standard concertina entanglement

Table F3 (Cont'd).

Label	Purpose
53	Computes values for a GPBTO
55	Computes number of 2-1/2-ton truck loads required
61	Outputs logistical requirements
63	Outputs number of long pickets needed
65	Outputs number of medium pickets needed
67	Outputs number of short pickets needed
69	Outputs number of barbed wire reels needed
71	Outputs cases of barbed tape needed
72	Outputs various logistical requirements
73	Outputs rolls of concertina needed
75	Outputs number of staples required
77	Outputs number of GPBTO units needed
79	Advises of program end and clears flags
Local Subroutines	
81	Inputs number of belts
82	Determines if wire use is tactical
83	Determines if wire use is protective
86	Determines man-hours required
87	Creates/stores experienced troop factor
88	Creates/stores driven picket factor (i.e., wood or U-shaped pickets)
89	Creates/stores night work factor

APPENDIX G:
GLOBAL UTILITY SUBROUTINES

This appendix provides detailed information about each of the global utility subroutines stored on the MILENG1/UTIL module. Figure G1 is a complete listing of these subroutines. Note that the global subroutines share many of the subordinate subroutines, and some even use other global subroutines in their entirety.

The global subroutines are used by each of six main application programs also stored on MILENG1/UTIL and can be used by field troops who may write their own programs. The subroutines can also be used by other military engineering programs that may one day be stored on future ROMs designed to be used only when the MILENG1/UTIL ROM is plugged in concurrently. This convention will save a lot of room on future ROMs and in user-developed programs -- room that would otherwise be needed to store similar subroutines. If field users adopt these subroutines, personnel throughout the military engineer community will find it easier to understand each others' programs.

Conventions

The following information on register use applies to all global utility subroutines. Registers 20 through 29 are reserved for existing and future global utility subroutines. Registers 20 through 23 are used to store temporarily up to 24 characters of a message that will be presented as a prompt of some sort to the program user. Each register stores up to six characters. Register 24 is an indirect storage register for a "pointer" to show where the next input value will be stored. Registers 25 and 26 store the minimum and maximum limits, respectively, of the current input variable. Registers 27 through 29 are reserved for global subroutines that may be developed in the future.

The following flag conventions are also used. Flag "10" records the results of a yes/no question. The flag is set if the response is "Y" (yes) and cleared if the response is "N" (no). Flag "9" is programmed into subroutine *I (numeric input) and *A (alpha input) to allow an answer already stored somewhere to be used instead of the user-defined values (alpha or numeric) that usually result when the *I and *A subroutines are invoked. Although this option is not used in any of the six main application programs on MILENG1/UTIL, the potential to bypass the normal user-input route and to use a value already in the system is available in these two sub-routines. This flexibility may be useful at times in future applications.

Flag "8" was not used in the MILENG1/UTIL programs. It should be reserved for future use as a "global use" flag, which "jumps over" certain parts of a program that do not have to be executed each time on repetitive runs. Flags "0" through "7" are reserved for application program use and are cleared each time the *F subroutine is invoked.

Global Subroutine *S

Running global subroutine *S should always be one of the first steps in any program; this insures that adequate data registers have been allocated for the program being executed. If enough registers are available, the subroutine returns to the main program and continues execution; if not, the user is prompted to resize the memory. The program must then be restarted.

Before using the subroutine, the entry condition must first be satisfied. The number of data registers required for the program is first loaded into the X register. Note that this is the actual number required, not the number of the highest register, which would be one less because register 00 counts as one register. The user then executes *S.

The following exit conditions are observed: *S returns to the calling program if adequate registers are available; otherwise, *S prompts the user to resize. One should assume that the stack contents are destroyed upon the return.

A sample calling sequence follows:

Program Instruction	Explanation
61	Loads "61" into X register; says that 61 registers (0 through 60) are required for this particular program;
XEQ "*S"	Calls the *S subroutine
xxxxxxxxxx	Resume with main program

Figure G2 is a detailed flowchart of *S. Note that *S also uses global subroutines *O and *D.

Global Subroutine *F

This subroutine clears flags "00" through "07"; it should be used to initialize a program and to "clean up" at the end of each application program.

To use the subroutine, the programmer simply executes *F. *F returns to the calling program once flags "00" through "07" have been cleared.

A sample calling sequence follows:

Program Instruction	Explanation
XEQ *F	Calls the *F subroutine
xxxxx	Resume with main program

Figure G3 is a detailed flowchart of *F.

Global Subroutine *I

Global subroutine *I prompts for numeric input; a tone alerts the user that input is required. If the input provided is not numeric, the user is prompted again.

The input value must pass a range check. If the input is out of range, the user is informed of the maximum or minimum acceptable value and reprompted. Global subroutine *I has a built-in option; if the user presses only the R/S key, he is prompted with the current value.

To use *I, the indirect register pointer in register 24 must be set to the data register the input is to be stored in. This pointer is incremented during each input call and has to be set only once if sequential input is to be stored in sequential registers. A prompt line and maximum and minimum acceptable value for the input are passed to the subroutine from the main program. The subroutine will not return to the main program until an acceptable value has been input. A "=?" is automatically appended to the prompt. If flag "9" is set, an "=" sign, the current value in the specified register, and a "?" are added to the prompt and displayed to the user.

Entry conditions when calling *I are as follows: register X must contain the minimum acceptable value, register Y must contain the maximum acceptable value, register A must contain the prompt. Register 24 must contain the address of the register that the input is to be stored in.

The exit conditions from *I occur when an acceptable value has been entered; *I then returns to the calling program. (If flag "9" is set and R/S is pressed without numeric entry, the current value is used.) The input value is stored in the specified register and in the X register. The rest of the stack should be considered destroyed.

A sample calling sequence follows:

Program Instruction	Explanation
SF 09	Optional: used if "current value" of variable is to be presented to user for verification.
30	Identifies register address where input is to be stored.
STO 24	Stores indirect register address in register 24.
123	Specifies maximum acceptable input value
ENTER	Places maximum value in Y register
-37	Specifies minimum acceptable input value and puts in X register
"HEIGHT"	Specifies prompt to be presented to user
XEQ "*I"	Calls subroutine *I

CF 09	Used only if SF 09 option is used
xxxxxxxx	Resume with main program

Note that if input is sequential, only the first two instructions have to be programmed before the first variable is input. Also, if the current value option is used, make sure the calculator is FIX'd to the desired setting before calling *I. A detailed flowchart of *I is presented in Figure G4. Note that *I also uses global subroutines *O and *D.

Global Subroutine *O

This subroutine displays labeled output. A label is passed to the subroutine, and an "=" and the value in the X register are appended. A two-tone sequence alerts the user to the output. The routine then displays the labeled answer. If a printer is attached, the output display is printed, and the program continues execution after a pause. If no printer is attached, program execution stops until the the R/S key is pressed.

Before executing the subroutine, the user must ensure that register X contains the numeric data to be displayed and that register A contains the label to be appended. When the subroutine is done, it returns to the calling program. No registers are affected.

A sample calling sequence follows:

Program Instruction	Explanation
4.27	Puts value to be displayed in X register
"ANSWER"	Puts label to be used in alpha register
XEQ "*O"	Executes the *O subroutine
xxxxxxxx	Resume with main program

Note that the calculator should be FIX'd to the desired accuracy before executing *O. A detailed flowchart for *O is shown in Figure G5. Note that *O also uses global subroutine *D.

Global Subroutine *D

This subroutine displays an alphanumeric text line. A two-tone sequence is used to alert the user; the routine then displays the contents of the alpha register. If a printer is attached, the output is printed and displayed; the program continues execution after a pause. If no printer is attached, program execution stops until the R/S key is pressed.

Before executing *D, the user must insure that register A contains the alphanumeric text to be displayed. When *D is done, it returns to the calling program. No registers are affected.

An example calling sequence follows:

Program Instructions	Explanation
"SAMPLE"	Puts text to be displayed in alpha register
XEQ "*D"	Execute the *D subroutine
xxxxxxxx	Resume with main program

Figure G6 is a detailed flowchart for *D.

Global Subroutine *Y

This subroutine takes a prompt that is passed to it and appends "(Y/N)?" to it. The operator must then respond with "Y" or "N" or the routine will reprompt. The answer to the query is returned to the calling program as flag "10" status. For "Y" responses, flag "10" is set; for "N" responses, it is cleared. The routine automatically places the calculator in the alpha mode before prompting and turns off the alpha mode after the user inputs a response.

Before executing *Y, the user must insure that register A contains the query text. When *Y is done, it returns to the calling program. Flag "10" status is affected.

A sample calling sequence follows:

Program Instructions	Explanation
"PRINT"	Puts query text in alpha register
XEQ "*Y"	Executes the *Y subroutine
FS? 10	Tests response: set=yes; clear=no
xxxxxxxx	Resume with main program

A detailed flowchart for *Y is shown in Figure G7.

Global Subroutine *A

This subroutine prompts the user for alpha input. A maximum of 12 alpha characters are stored. A tone sounds to alert the operator that input is required. A built-in option allows the programmer to prompt the user with the "current value" of the alpha variable. Under this option, if R/S is pressed, the "current value" of the text will be used when the program continues execution.

This subroutine requires that the indirect register address (pointer) stored in register 24 be set to the data register the alpha input is to be stored in. The alpha input is stored in two registers, six characters in each. The pointer in register 24 is incremented twice during each input call, so it only has to be set once if a string of input is to be put in sequential registers. A prompt line to label the input is passed to the subroutine, and a "=?" is added to this prompt by the subroutine.

To use the subroutine, the entry conditions must be satisfied. Register A must contain the prompt. Register 24 must contain the address of the register that the first six characters of the input will be stored in. If the programmer sets flag "9" before calling *A, the "current value" in the specified registers are appended to the prompt and are presented to the user for verification. If the user agrees with the alpha value assigned, he presses the <u>R/S</u> key, and the program uses that alpha value for the variable. If the user elects to change the value, he simply enters the correct alpha string (12 characters or fewer), and presses the <u>R/S</u> key. This new alpha value would then be used in place of the "current value."

The following exit conditions result: *A returns to the calling program, and the input value is stored in the specified registers. No other registers are affected.

A sample calling sequence follows:

Program Instructions	Explanation
SF 09	(Optional) If current value of variable is to be presented to user
30	First register address where input is to be stored
STO 24	Stores first indirect register address in register 24
"VARIABLENAME"	Specifies prompt to be presented to user
XEQ "*A"	Calls the "*A" subroutine
CF 09	Used only with "SF 09" option.
xxxxxxxx	Resume with main program.

Note that if input is sequential, only the second and third instructions need to be programmed before the first variable is input. A detailed flowchart is shown in Figure G8.

Global Subroutine *C

This subroutine clears a specified range of registers by storing a "0" in it. This subroutine should be used instead of CLRG so that the contents of registers not used in the application program are preserved.

Before the subroutine can be used, the entry conditions must be satisfied. Register X must contain the range of registers to be cleared; the format is fff.lll, where fff is the address of the first register to be cleared and lll is the address of the last register to be cleared.

After the specified registers have been cleared, *C exits to the calling program. The stack should be considered destroyed.

A sample calling sequence follows:

Program Instructions	Explanation
30.045	Specifies address of registers to be cleared (30 through 45)
XEQ "*C"	Calls the *C subroutine
xxxxxxxx	Resume with main program

A detailed flowchart for *C is shown in Figure G9.

Global Subroutine *R

This subroutine "rounds-up" a value and displays the integer portion of the number. The subroutine first adds 0.99 to the value stored in register X, then uses the integer portion of that value. Before entering the subroutine, the user must insure that register X contains the value to be rounded. When *R is done, it returns to the calling program.

A sample calling sequence follows:

Program Instructions	Explanation
5.49	Specifies value to be rounded; could also be a recall RCL instruction
XEQ *R	Calls the *R subroutine
xxxxxxxx	Resume with main program

Figure G10 is a detailed flowchart for *R.

Global Subroutine *P

This subroutine displays the "END PROGRAM" message in large type. A two-tone sequence sounds to alert the user; *P then displays the message. There are no pre-entry conditions for *P. The subroutine is called directly with the instruction, "XEQ *P". When *P is done, it returns to the calling program; no registers are affected. A detailed flowchart of *P is in Figure G11.

```
♦LBL "*S"
"RESIZE>"
1
-
SF 25
RCL IND X
FS?C 25
RTN
FIX 0
1
+
XEQ "*O"
STOP

♦LBL "*I"
XEQ 13
STO 25
RDN
STO 26

♦LBL 08
FS? 09
RCL IND 24
"⊢="
FS? 09
ARCL IND 24
"⊢?"
CF 22
TONE 5
AVIEW
STOP
FS? 22
GTO 01
FC? 09
GTO 00
ISG 24
STO X
RTN

♦LBL 00
XEQ 14
GTO 08

♦LBL 01
"MUST BE"
RCL 25
X<=Y?
GTO 02
"⊢>"

♦LBL 03
XEQ "*O"
GTO 00

♦LBL 02
"⊢<"
RDN
RCL 26
X<Y?
GTO 03
RDN
STO IND 24
ISG 24
STO X
RTN

♦LBL "*O"
"⊢="
ARCL X

♦LBL "*D"
TONE 8
TONE 9
AVIEW
FS? 55
PSE
FC? 55
STOP
RTN

♦LBL "*Y"
XEQ 13

♦LBL 10
"⊢(Y/N)"
SF 10
XEQ 05
ASTO X
"Y"
ASTO Y
X=Y?
RTN
CF 10
"N"
ASTO Y
X=Y?
RTN
XEQ 14
GTO 10

♦LBL "*A"
XEQ 13

♦LBL 09
CF 10
"⊢="
FC? 09
GTO 05
ARCL IND 24
ISG 24
STO X
ARCL IND 24
DSE 24

♦LBL 05
"⊢?"
CF 23
AON
TONE 5
AVIEW
STOP
AOFF
FS? 10
RTN
FS? 23
GTO 07
FS? 09
GTO 06
XEQ 14
GTO 09

♦LBL 07
ASTO IND 24
ASHF
ISG 24
STO X
ASTO IND 24
DSE 24

♦LBL 06
CLA
ARCL IND 24
ISG 24
STO X
ARCL IND 24
ISG 24
STO X
RTN
```

Figure G1. Global subroutines -- listing.

```
134♦LBL 13
135 ASTO 20
136 ASHF
137 ASTO 21
138 ASHF
139 ASTO 22
140 ASHF
141 ASTO 23

142♦LBL 14
143 CLA
144 ARCL 20
145 ARCL 21
146 ARCL 22
147 ARCL 23
148 RTN
149 "26/01"
150 .END.

01♦LBL "*F"
02 7

03♦LBL 04
04 CF IND X
05 DSE X
06 GTO 04
07 CF 00
08 RTN

09♦LBL "*C"
10 0

11♦LBL 11
12 STO IND Y
13 ISG Y
14 GTO 11
15 RTN

16♦LBL "*R"
17 .99
18 +
19 INT
20 RTN

21♦LBL "*P"
22 SF 12
23 "END PROGRAM"
24 TONE 8
25 TONE 9
26 AVIEW
27 CF 12
28 ADV
29 ADV
30 ADV
31 ADV
32 RTN
33 "NOTE:CERL VERSI"
34 "⊢ON,2APR82"
35 .END.
```

Figure G1. (Cont´d).

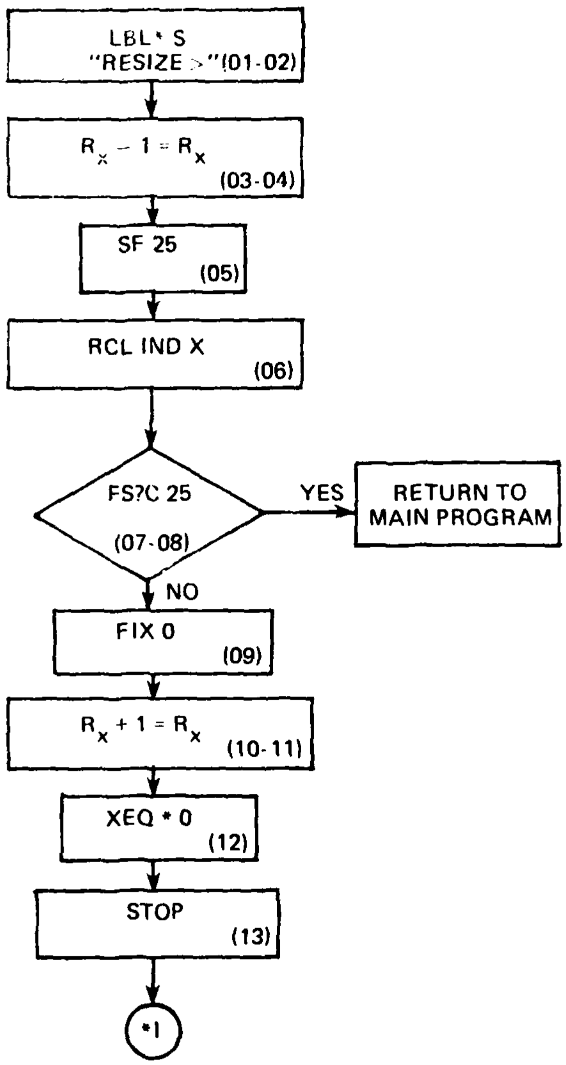


Figure G2. Detailed program flowchart for *S subroutine.

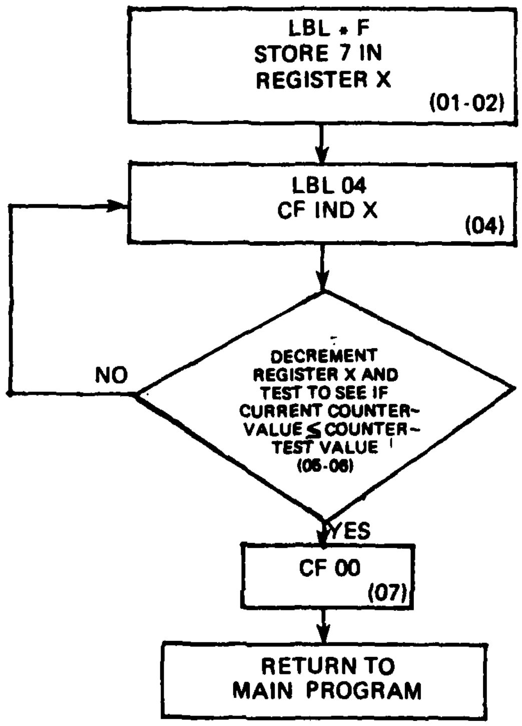


Figure G3. Detailed program flowchart for *F subroutine.

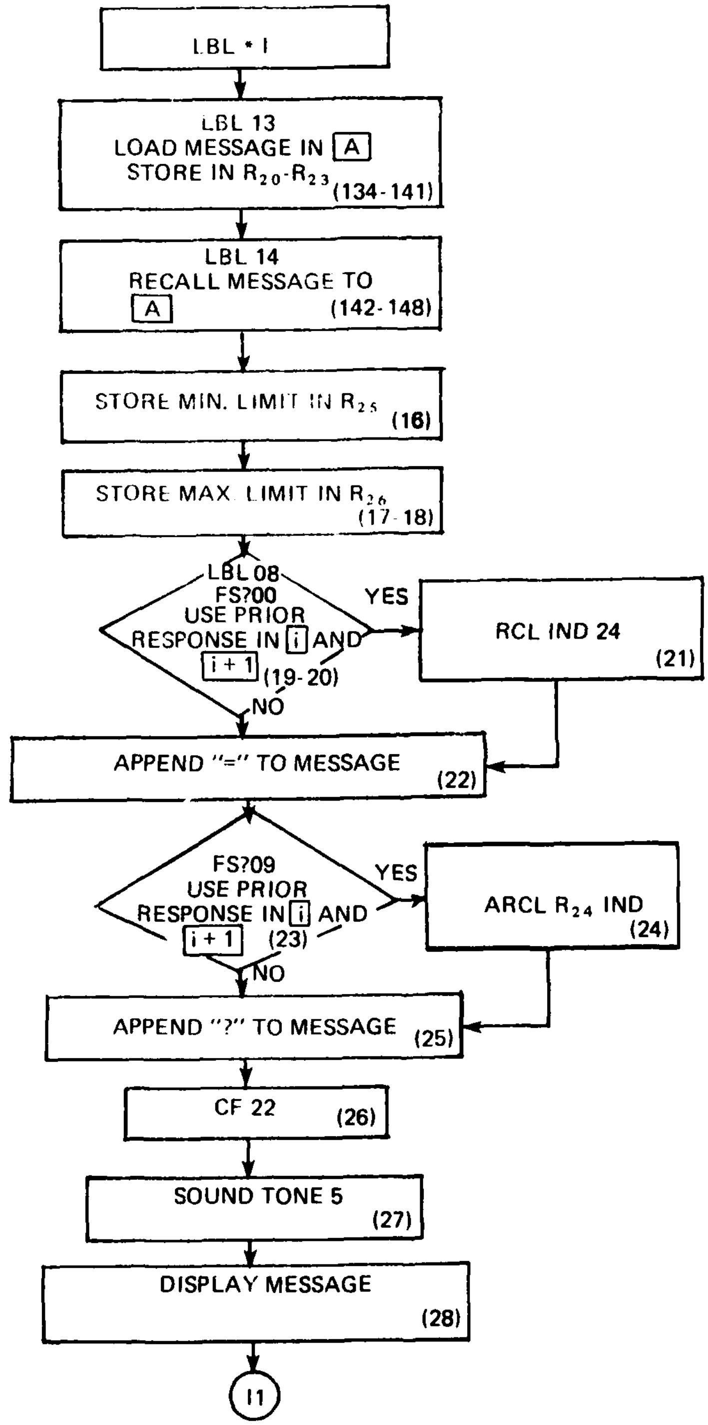


Figure G4. Detailed program flowchart for *I subroutine.

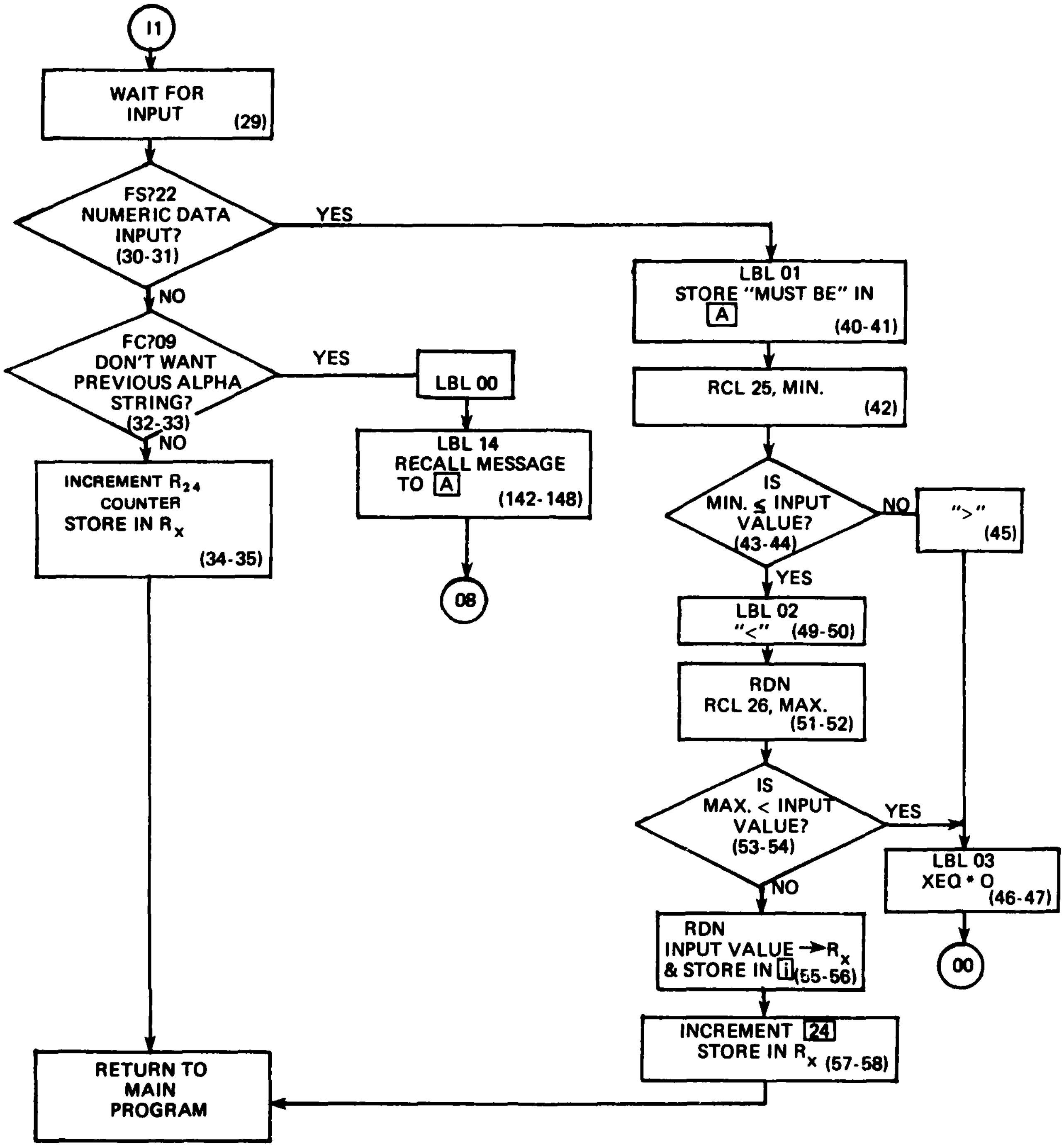


Figure G4. (Cont'd).

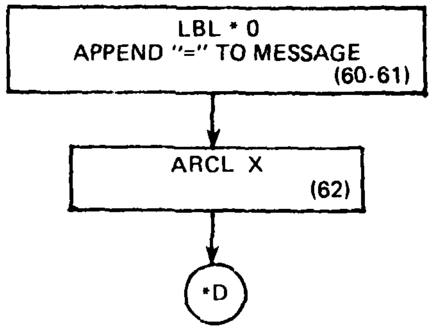


Figure G5. Detailed program flowchart for *0 subroutine.

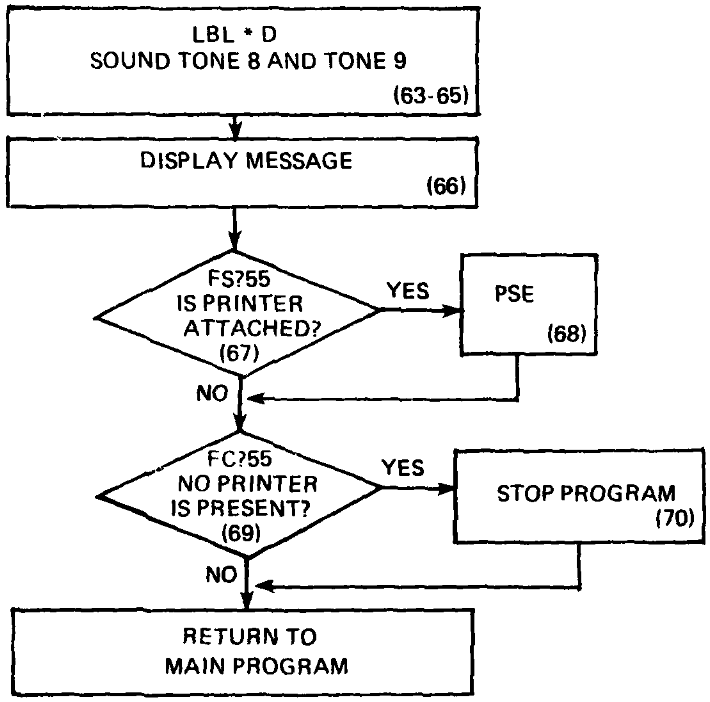


Figure G6. Detailed program flowchart for *D subroutine.

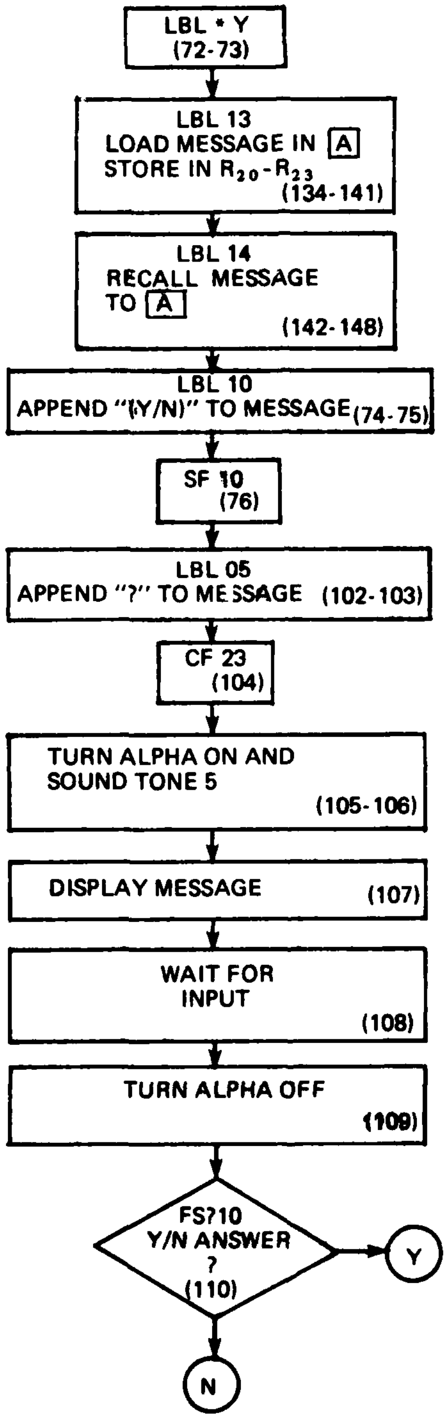


Figure G7. Detailed program flowchart for *Y subroutine.

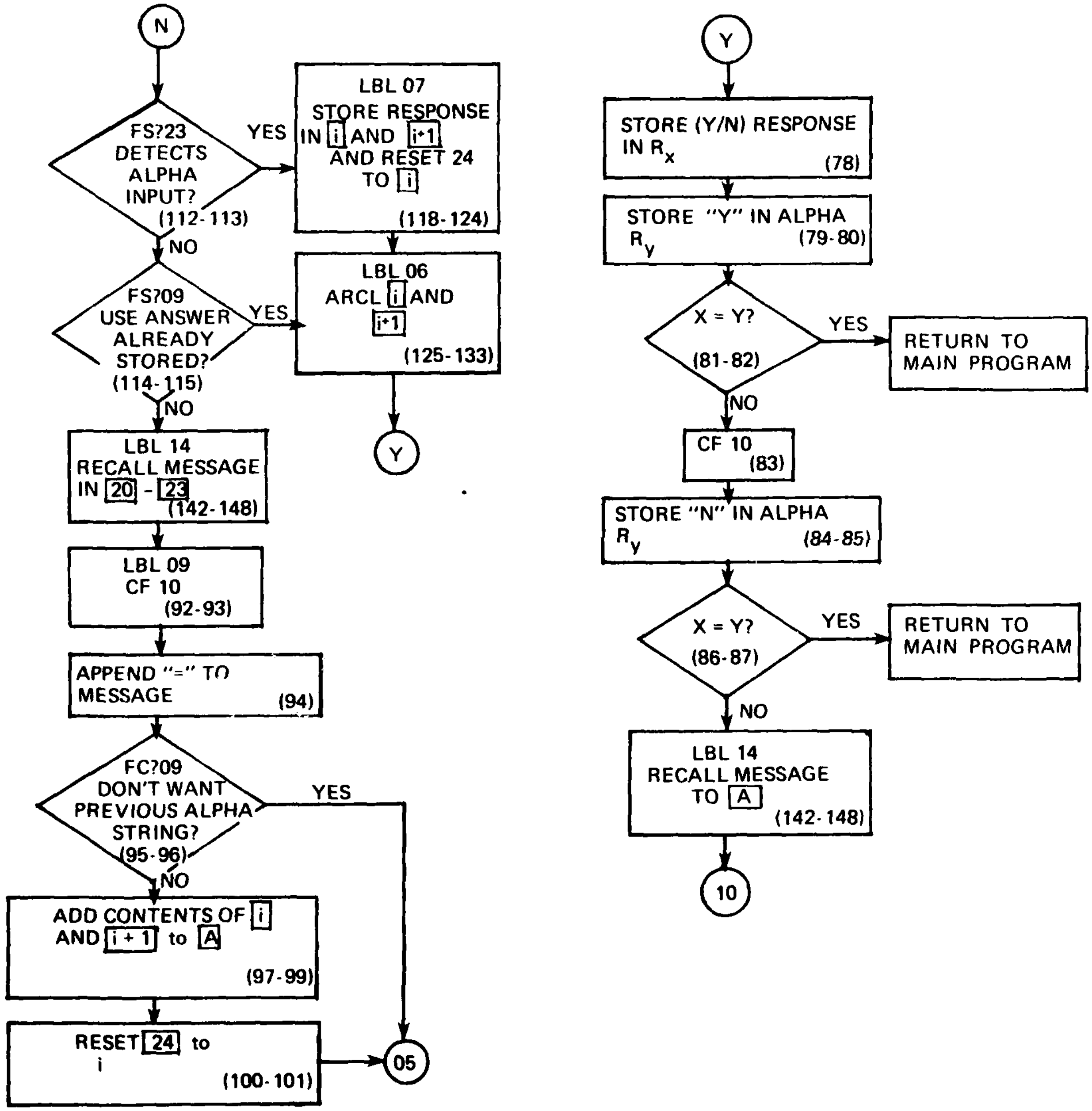


Figure G7. (Cont'd).

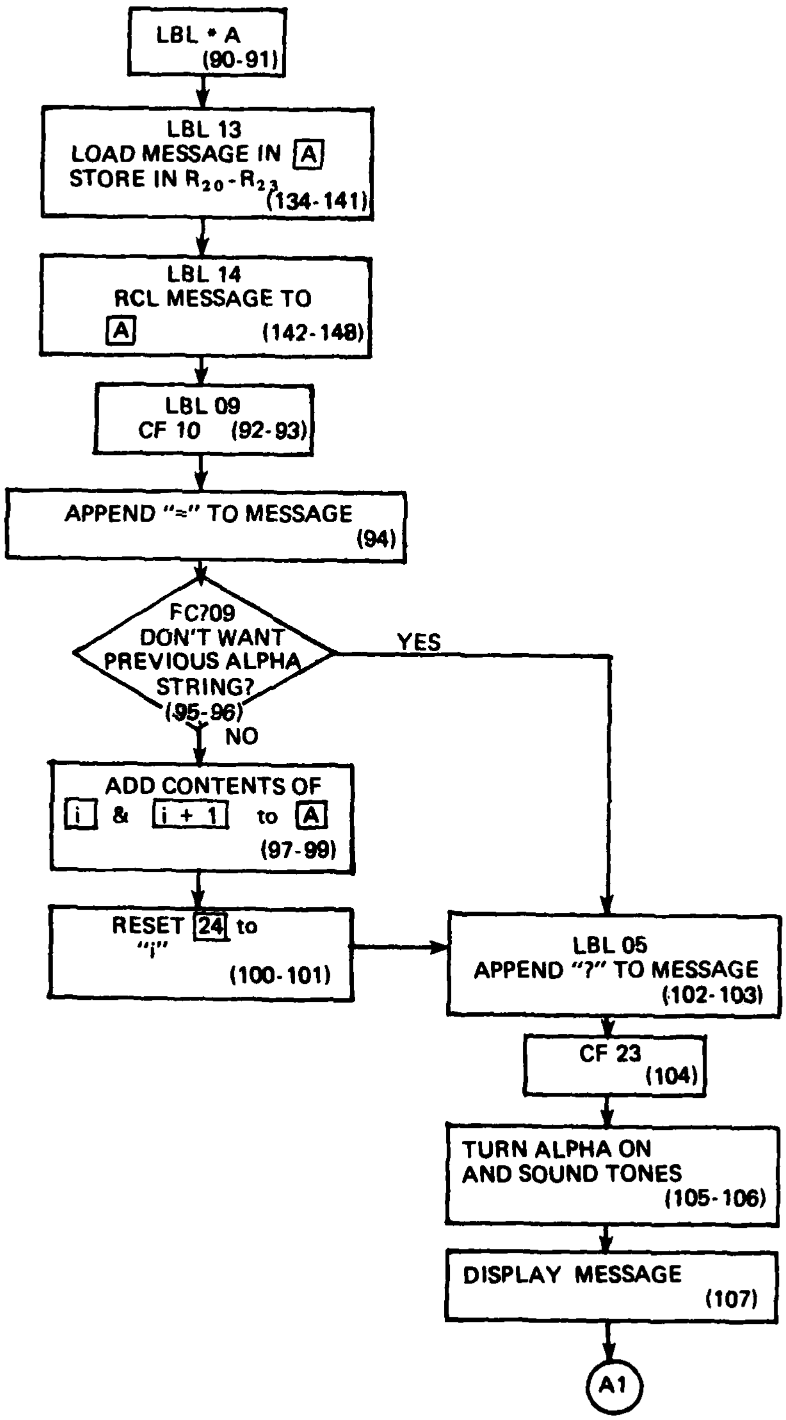


Figure G8. Detailed program flowchart for *A subroutine.

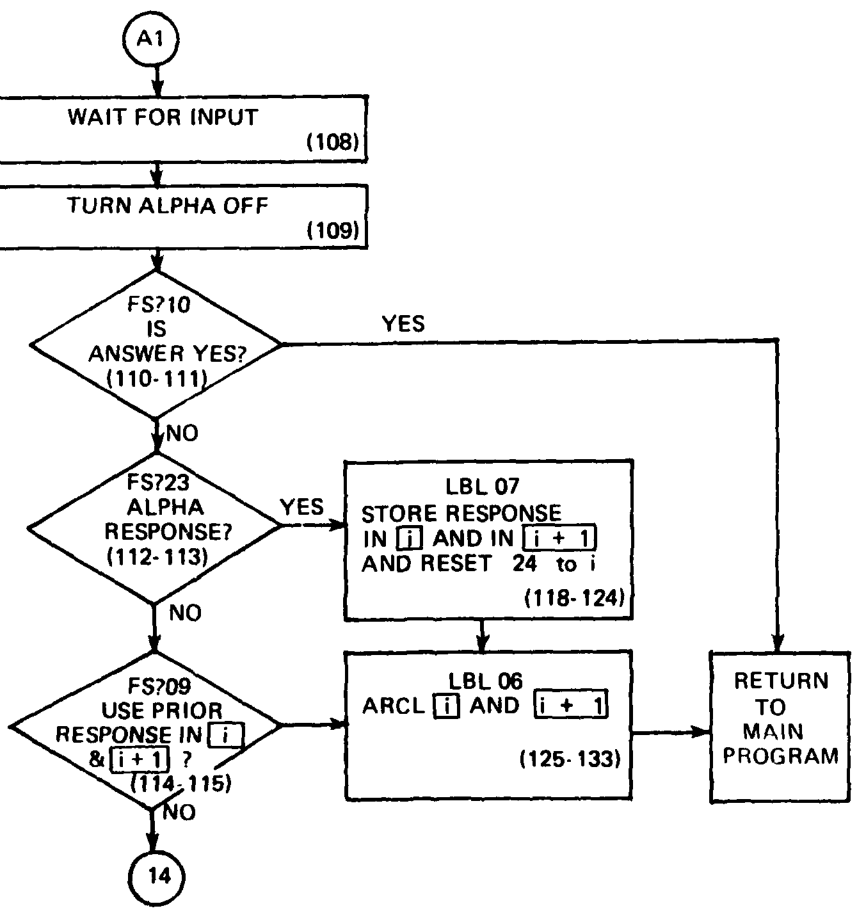


Figure G8. (Cont'd).

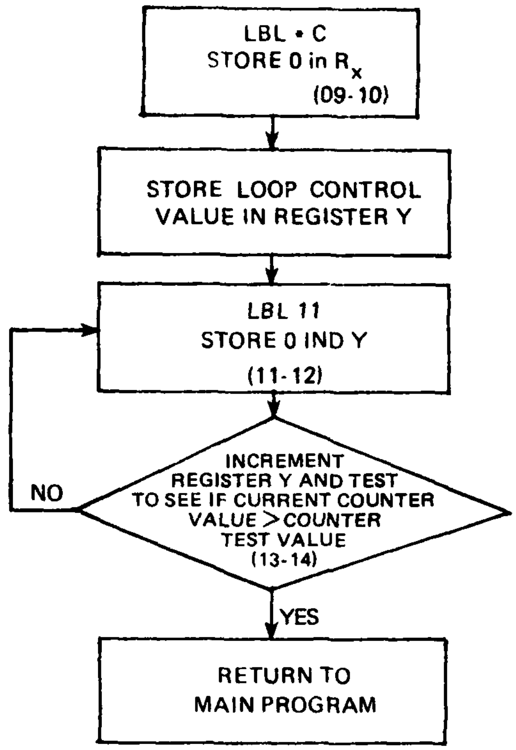


Figure G9. Detailed program flowchart for *C subroutine.

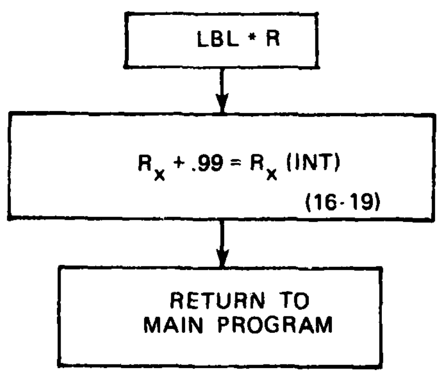


Figure G10. Detailed program flowchart for *R subroutine.

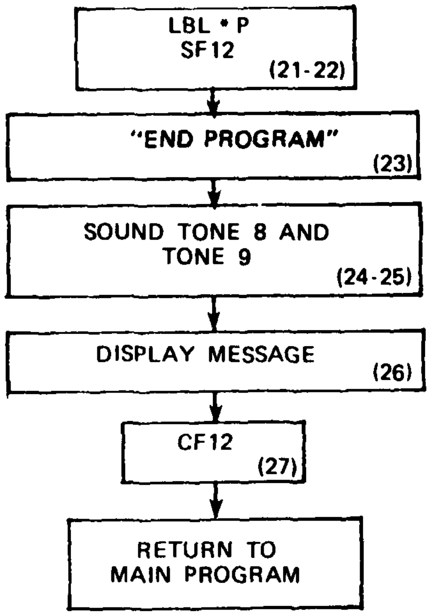


Figure G11. Detailed program flowchart for *P subroutine.

MILITARY ENGINEERING PROGRAM DOCUMENTATION

PROGRAM TITLE:

AUTHOR(S) ______________________ DATE ______________

ADDRESS ______________________ TEL. NO. ______________

CITY ______________ STATE ______________ ZIP ______________

POINT OF CONTACT: ______________________ ______________

ADDRESS ______________________ TEL. NO. ______________

CITY ______________ STATE ______________ ZIP ______________

DOCUMENTATION CHECKLIST

- ☐ PROGRAM ABSTRACT
- ☐ PROGRAM SEQUENCE
- ☐ EXAMPLE PROBLEM(S) – WITHOUT PRINTER
- ☐ EXAMPLE PROBLEMS(S) – WITH PRINTER
- ☐ GENERAL PROGRAM INFORMATION
- ☐ PROGRAM METHODOLOGY
- ☐ PROGRAM LABELS
- ☐ PROGRAM FLAGS
- ☐ PROGRAM REGISTERS
- ☐ PROGRAM "WIRING DIAGRAM"
- ☐ PROGRAM LISTING
- ☐ DETAILED PROGRAM FLOWCHART
- ☐ MAGNETIC CARD COPY OF PROGRAM

PROGRAM ABSTRACT

PROGRAM TITLE:

ABSTRACT:

ACCESSORIES REQUIRED:

REGISTER ALLOCATIONS:

DATA REGISTERS REQUIRED ____________ (EXECUTE "SIZE _ _ _ ")

PROGRAM REGISTERS REQUIRED ______

Page of

PROGRAM SEQUENCE

PROGRAM TITLE:

☐ CONTINUATION SHEET USED

Page of

EXAMPLE PROBLEM (WITHOUT PRINTER)

PROGRAM TITLE:

STATEMENT OF EXAMPLE PROBLEM:

STEP	PRESS	RESULTING DISPLAY	COMMENTS

☐ CONTINUATION SHEET USED

Page of

EXAMPLE PROBLEM CONTINUATION SHEET (WITHOUT PRINTER)

PROGRAM TITLE:

STEP	PRESS	RESULTING DISPLAY	COMMENTS

☐ CONTINUATION SHEET USED

Page of

EXAMPLE PROBLEM (WITH PRINTER)

PROGRAM TITLE:	
STATEMENT OF EXAMPLE PROBLEM:	STATEMENT OF EXAMPLE PROBLEM:

☐ CONTINUATION SHEET USED

Page of

GENERAL PROGRAM INFORMATION

PROGRAM TITLE:

ABBREVIATIONS:

OPERATING LIMITS AND WARNINGS:

SOURCE REFERENCE NOTES:

Page of

PROGRAM METHODOLOGY

PROGRAM TITLE:

☐ CONTINUATION SHEET USED

Page of

PROGRAM LABELS

PROGRAM TITLE:	
LABEL	DESCRIPTION

☐ CONTINUATION SHEET USED

Page of

PROGRAM FLAGS

PROGRAM TITLE:	
FLAG	DESCRIPTION

☐ CONTINUATION SHEET USED

Page of

PROGRAM REGISTERS

PROGRAM TITLE:	
REGISTER	DESCRIPTION

☐ CONTINUATION SHEET USED

Page of

PROGRAM "WIRING DIAGRAM"

PROGRAM TITLE:

☐ **CONTINUATION SHEET USED**

DETAILED PROGRAM FLOWCHART

PROGRAM TITLE:

☐ CONTINUATION SHEET USED

Page of

CERL DISTRIBUTION

Chief of Engineers
ATTN: Tech Monitor
ATTN: DAEN-ASI-L (2)
ATTN: DAEN-ZCM

FESA, ATTN: Library 22060

416th Engineer Command 60623
ATTN: Facilities Engineer

ROK/US Combined Forces Command 96301
ATTN: EUSA-HHC-CFC/Engr

US Military Academy 10996
ATTN: Dept of Geography &
Computer Science

Engineer Studies Center 20315
ATTN: Library

AMMRC, ATTN: DRXMR-WE 02172

USA ARRCOM 61299
ATTN: DRCIS-RI-I
ATTN: DRCIS-IS

DLA, ATTN: DLA-WI 22314

FORSCOM
FORSCOM Engineer, ATTN: AFEN-FE

Fort Belvoir, VA 22060
ATTN: ATZA-DTE-EM
ATTN: ATZA-DTE-SW
ATTN: ATZA-FE
ATTN: Engineer Library
ATTN: Canadian Liaison Officer (2)
ATTN: IWR Library

Cold Regions Research Engr Lab 03755
ATTN: Library

ETL, ATTN: Library 22060

Waterways Experiment Station 39180
ATTN: Library

HQ, XVIII Airborne Corps & Ft Bragg 28307
ATTN: Library

Defense Technical Info Center 22314
ATTN: DDA (12)

US Government Printing 22304
Receiving Section/Depository Copies (2)

US Army, Commander
ATTN: 7 Engr Bde 09154
ATTN: 18 Engr Bde 09164
ATTN: 20 Engr Bde 28307
ATTN: 130 Engr Bde 09165
ATTN: 2 Engr Grp 96301
ATTN: 36 Engr Grp 31905
ATTN: 937 Engr Grp 66442
ATTN: 1 Engr Bn 66442
ATTN: 2 Engr Bn 96224
ATTN: 3 Engr Bn 31313
ATTN: 4 Engr Bn 80913
ATTN: 5 Engr Bn 65473
ATTN: 7 Engr Bn 71459
ATTN: 8 Engr Bn 76545
ATTN: 9 Engr Bn 09162
ATTN: 10 Engr Bn 09701
ATTN: 11 Engr Bn 22060
ATTN: 12 Engr Bn 09111
ATTN: 13 Engr Bn 93941
ATTN: 14 Engr Bn 93941
ATTN: 15 Engr Bn 98433
ATTN: 16 Engr Bn 09696
ATTN: 17 Engr Bn 76546
ATTN: 19 Engr Bn 40121
ATTN: 20 Engr Bn 42223
ATTN: 23 Engr Bn 09165
ATTN: 27 Engr Bn 28307
ATTN: 30 Engr Bn 22060
ATTN: 34 Engr Bn 66442
ATTN: 39 Engr Bn 01433
ATTN: 43 Engr Bn 31905
ATTN: 44 Engr Bn 96483
ATTN: 46 Engr Bn 36362
ATTN: 52 Engr Bn 80913
ATTN: 54 Engr Bn 09026
ATTN: 62 Engr Bn 76544
ATTN: 65 Engr Bn 96857
ATTN: 76 Engr Bn 20755
ATTN: 78 Engr Bn 09351
ATTN: 79 Engr Bn 09360
ATTN: 82 Engr Bn 09139
ATTN: 84 Engr Bn 96857
ATTN: 92 Engr Bn 31313
ATTN: 94 Engr Bn 09175
ATTN: 237 Engr Bn 09176
ATTN: 249 Engr Bn 09360
ATTN: 293 Engr Bn 09034
ATTN: 299 Engr Bn 73503
ATTN: 307 Engr Bn 28307
ATTN: 317 Engr Bn 09757
ATTN: 326 Engr Bn 42223
ATTN: 547 Engr Bn 09175
ATTN: 548 Engr Bn 28307
ATTN: 549 Engr Bn 28307
ATTN: 549 Engr Bn 09081
ATTN: 559 Engr Bn 09165
ATTN: 563 Engr Bn 09154
ATTN: 565 Engr Bn 09164
ATTN: 588 Engr Bn 71459
ATTN: 649 Engr Bn 09081
ATTN: 652 Engr Bn 96858
ATTN: 802 Engr Bn 96271
ATTN: 864 Engr Bn 98433

National Guard Bureau 20310
Installation Division

Thomas, Laure A
Software documentation for MILENG1/UTIL read only memory module / by Laure A. Thomas, John M. Deponai III. -- Champaign, Ill : Construction Engineering Research Laboratory ; available from NTIS, 1982.
174 p. (Technical report / Construction Engineering Research Laboratory ; P-134)

1. Military engineering 2. Programmable calculators 3. HP41c
I. Deponai, John M. II. Title. III. Series: Technical report (Construction Engineering Research Laboratory) ; P-134.

LME

-8